Mobile Application Primer
6.1 Mobile, an opportunity to focus on the essentials .................................68
6.2 From Mobile app to mobile web .............................................................68
6.3 User centred approach ........................................................................68
6.4 Surveys ...................................................................................................69
6.5 Prototyping before production ...............................................................69
6.6 Don’t underestimate the systems integration effort ...............................69
6.7 Buy versus Build ..................................................................................70

Appendix I: UCC Student Survey Data ......................................................71

Appendix II: Buying in, not building: an alternative approach to
development .............................................................................................77

Appendix III: Mobile app candidates .........................................................79

Appendix IV: UCC prototype technical architecture .................................91
1. Introduction

“The project described below falls within HEAnet’s Mobility Strategy. This document is one of a number of deliverables to come from the collaborative Mobile Broadband agreement between HEAnet and O2, signed in September 2011. Given the significant proliferation of mobile devices and the need for our clients to provide mobile app services - this document sets out to help them provide mobile apps that meet the needs of a variety of end-user communities.”

Peter O’Halloran, Head of Client Services & Operations, HEAnet

Two things happened that made most organisations seriously consider their mobile strategy:

1. the continuing growth of smartphone adoption around the world, and
2. the trends that show that, by 2015, more internet users will access the Internet through mobile devices rather than PCs¹.

With HEAnet’s remit to maintaining the most cost-effective and technically advanced solutions to its user community, “mobile” is an area that requires both a detailed understanding and a strategic approach.

The objective of this project was to produce a prototype mobile application for UCC while also to document the design & development to assist other HEAnet clients who are considering embarking on their own mobile application projects.

¹ http://www.idc.com/getdoc.jsp?containerId=prUS23028711
1.1 Acknowledgments

This document was produced by iQ Content, working together with Telefonica, HEAnet and University College Cork. We’d like to thank all parties involved and in particular, University College Cork IT Services, who gave up their time and resources for the case study.

The project ran from May 2012 to October 2012.

1.2 How this document is structured

1. A brief look at the mobile landscape
2. A user-centred design approach to mobile
3. A case study with UCC, applying the user-centred design approach to build a prototype
4. Key lessons learned
5. Appendix I: UCC student survey data
6. Appendix II: Buying in, not building: alternative approach to development
7. Appendix III: Mobile app candidates.
8. Appendix IV: Prototype technical architecture

1.3 Summary of approach

To design and build a successful app, you need to identify

1. **Who**: the users for your app
2. **What**: what is it the app should do for them
3. **How**: the technical implementation

This report outlines how we identified each of the above and the user centred design approach we took while working with UCC.

In this document we cover the following:

1. Identify the audience(s) for your mobile app.
2. Explore the opportunities that smartphone capabilities present and what your audience needs most.
3. Conduct research on what services or features to prioritise.
4. Design the experience you want your users to have.
5. Decide what technology approach you should take.
6. Considerations once the app is live.
2. The mobile landscape

2.1 Some global trends

Mobile has exploded. From the device side, the combined installed base of smartphones will surpass 1.82 billion units by 2013, eclipsing the total of 1.78 billion PCs. By 2015, more internet users will access the Internet through mobile devices than through PCs or other wired devices.

From the adoption and behaviour side, a good indicator of the growth is mcommerce.

PayPal initially reported the following, with the last figure adjusted because of even faster than expected growth in 2012.

- $25 million in ‘08
- $141 million in ‘09
- $750 million 2010/2011
- $4 billion in 2011
- $10 billion in 2012

As for another well known site, Facebook, 50% of their total audience (845 million) uses the service on mobile devices (425 million).

---

3 http://www.idc.com/getdoc.jsp?containerId=prUS23028711
4 http://www.mobilecommercedaily.com/2011/03/22/paypal-mobile-transactions-exceed-6m-per-day-ctia-keynote
6 https://developers.facebook.com/blog/post/2012/02/27/helping-improve-the-mobile-web/
2.2 Local trends

The massive growth of mobile is not confined to the global stage either. Looking at some more local sources

2.2.1 Mobile web traffic

Above, 1000% growth in website traffic via mobile devices over two years for a major Irish utility company. We're seeing the same pattern across the board, regardless of industry.

2.2.2 Wireless access on campus

On campus in UCD, growth in wireless versus wired access has accelerated rapidly, almost doubling since 2009.
2.2.3 Smartphone adoption

In their “Our Mobile Planet” report, Google report 43% smartphone penetration in Ireland\(^7\) (while we estimate smartphone adoption amongst the student population at 79%, see Appendix I). In terms of user behaviour, some more telling figures from the Google report:

- 61% have used their smartphones every day in the past 7 days
- 75% never leave home without it
- 60% search on their smartphones every day
- 89% of smartphone users have looked for local information; 90% of them took action as a result

2.2.4 Smartphone behaviour patterns

So, we can see from Google’s data above that smartphones are used everywhere. But what are they actually used for?

Google have categorised three patterns of mobile behaviour\(^8\):

1. Repetitive Now

\(^7\) [http://services.google.com/fh/files/blogs/our_mobile_planet_ireland_en.pdf](http://services.google.com/fh/files/blogs/our_mobile_planet_ireland_en.pdf)

2. Bored Now
3. Urgent Now

1. Repetitive now
The "repetitive now" user is looking for recurring real-time information, for example a student timetable or a café menu.

2. Bored now
The "bored now" user has time on their hands. They could be waiting for someone, commuting on public transport or sitting in the library or perhaps on the sofa at home. News or events could cater for this kind of user.

3. Urgent now
The user needs an answer to something now, often related to their current location. “Where is the nearest lab with free PCs?” or “where is the examination hall?” could be two good examples of the questions urgent now users need to answer.

A good mobile app could cater for one or all of the behaviour patterns - good apps often cater for all.

2.3 Three things to consider
1. Consider smartphone capabilities as an opportunity to innovate
2. Use mobile as an opportunity to focus on the essentials.
3. Don’t just focus on mobile.

2.3.1 Consider smartphone capabilities as an opportunity to innovate.
What makes mobile different? A number of things - smartphones are personal devices - far more personal than a desktop or laptop could ever be. We bring them everywhere with us. Smartphones have a whole host of capabilities that desktop or laptops just don’t have.

Each one of these capabilities provides unique opportunities to provide a new or better experience.

Here are just some of the capabilities on a modern smartphone.

- Device positioning & motion: from an accelerometer
- Gyroscope: 360 degrees of motion
- Location detection
- Multi-touch sensors
• Orientation & direction: direction from a digital compass
• Cameras: capture/input from a camera, often front & back
• Audio input & output
• Connectivity: through Bluetooth, Wifi & 3G
• Other sensors: light/dark environment awareness.
• NFC: Near Field Communications through RFID readers

**Example:** Location-detection (versus desktop experience)

Previously, on the web through the desktop, “smart” websites could tell where you were using reverse IP look-up. The closest they could get with any degree of accuracy was country.

Now, on a smartphone, you can be pinpointed to your exact location - helping a user in an unfamiliar location to answer questions like where’s my nearest café or Wifi hotspot.

So, with these mobile capabilities at your disposal, there are opportunities to exploit them to redefine existing or create new compelling use cases and experiences.

But, a word of caution, just because you can, doesn’t mean you should.

### 2.3.2 Use mobile as an opportunity to focus on the essentials.

With smaller screen size, you have some inherent constraints.

With a smaller screen size, you have a smaller canvas to work on, which forces focus & priority on what matters most. These constraints can be quite beneficial.

Other constraints like short battery life & slow networks mean that speed really matters - again, forcing or encouraging faster websites by reducing file sizes and optimising graphics using local caching (a HTML5 feature).

**2.3.3 Don’t just focus on mobile.**

While mobile is the focus of this document we know that mobile is just one device among others for our users.

“Losing 80% of your screen space forces you to focus. You need to make sure that what stays on the screen is the most important set of features for your customers and your business. There simply isn’t room for any interface debris or content of questionable value.”

Luke Wrobelski, author of “Mobile First”. 
Users will rightly expect to interact with various services and information across a wide variety of media - their phone, their laptop, a lab PC.

Getting your data and content right first means that you’re in a good position to deliver your content across multiple channels.
3. User centred design approach to mobile.

The key to ensuring the success of any app is to understand an organisation’s goals for it, what users need from it and how technology can enable it.

Great apps take all three into consideration, while ignoring any of them will ultimately result in an app that doesn’t deliver any benefits to anybody or is technically unfeasible.

The best way to achieve a useful and usable product is by adopting and applying a user-centred design process. User-centred design is based around engagement with and a focus on users throughout the design and development life-cycle.
Above, user-centred design can help bring products beyond functional and usable, to lovable.

This is opposed to a developer-centric approach, where the product is developed starting with the technology, and only later are users considered, by which time it’s often too late because the product represents an engineer’s “implementation model” rather than a user’s “mental model”.

**Phases of user-centred design**

The phases & some typical tasks within a user centred design approach are as follows:

1. **Discovery**: Research & analysis
2. **Design**: workshops, navigation and wireframes
3. **User testing**
4. **Visual design**
3.1 Discovery & analysis

The primary focus of the discovery phase is on the users – who are the users for the mobile app and what is it they need? The outputs of this crucial piece of analysis inform the rest of the process, including what to design and how to build it.

Some of the key activities in discovery and analysis are as follows:

**Stakeholder interviews**
Stakeholder interviews are an excellent tool for getting input into the design process. They’re also a great technique for securing buy-in early in a project, thus avoiding potential internal wrangles and politics further down the road.

Ultimately, these interviews are all about getting an understanding of the business context and the technical environment in which the mobile app could be built.

A stakeholder is anyone with responsibility for the product being designed - whether that’s the designer, the developer or the business owner who will realise a benefit or need to report on the return on investment (ROI) of the project.

**Online surveys & questionnaires**
Online surveys are a useful, efficient and cost-effective tool for gathering data from relatively large sample sizes.

They can contribute towards building a clear picture of your web audience, their opinions and their preferences so it’s not just quantitative data they can provide.

**User interviews**
User interviews give us the opportunity to find out about the goals and needs of our users. How, where and why would they use a university app? What do they do right now in the absence of a university app?

Interviews are one-to-one because we believe it is more productive to conduct small interviews rather group interviews, where a handful of voices often dominate the proceedings and voices (and insights) can easily be lost in the crowd. These interviews lasted no longer than an hour.

**Peer analysis**
Evaluation of the strengths and weaknesses of comparable offerings from peers or competitors

**Persona creation**
Personas are a brief representation of common user goals and behaviour. In most cases, they’re created from the synthesis of research activities including interviews, surveys and analytics. In addition to goals and behaviours, they also include some fictional personal details to help “personify” the design targets, making them realistic and relevant.

**Desk research**
Review of any existing research or previous literature relating to the project

---

**3.2 Design**

The following activities make up the design phase. The approach is based around generation of a number of ideas or directions before selecting one for refinement.

**Collaborative design workshops**
Working alongside stakeholders to produce concepts and generation (and prioritisation) of design ideas.

**Context scenarios**
Creation of narrative descriptions of the experiences we want our users to have with what we’re designing.

**Sketching**
A technique used to explore early designs or the essence of an idea. This approach helps make vague ideas more concrete while discovering alternative ideas. “Sketching” is not necessarily drawing, but an illustration, in whatever form, of the many possible designs. The idea that feels like the most promising is chosen to be further developed into a prototype.

**Prototyping**
A working model of the design to test ideas and designs early and iteratively, “getting the right design, before getting the design right”.

**Wireframes**
A visual blueprint. Wireframes are used to explore design alternatives as well as document the agreed final form and behaviour of an app or website.

**Visual design**
Bringing form to the function, adding branding, colour, icon and typography to create a polished look and feel.

We’ve called out two specific activities, context scenarios and prototyping because we think they’re vital activities involved in producing superior designs.
3.2.1 Spotlight on design: context scenarios

Writing context scenarios is a technique that uses the power of storytelling to communicate ideas. They’re essentially a tool for thinking, and anyone can use them.

In addition to communicating ideas, context scenarios or user stories are a very powerful method to imagine a better experience for our users when they interact with our designs.

Context scenarios shouldn’t represent too much by way of systems, i.e. the “how”, nor should be the written in the formal, passive voice of a detailed use case.

Instead, they should focus on the people - helping us to think beyond technology, features or functions and towards experiences.

The following list can help structure a context scenario:

1. Who would use this app?
2. Where could this person be when using it?
3. How will this app be used - on a mobile or desktop, or both?
4. What’s the ultimate goal of using this app?

Here are some example of some short context scenarios:

Mary is looking at the lecture timetable. Her first class in the next morning is in ORB 345. She doesn’t know which building is ORB so she uses the app to find out.

John just finished his revision lecture and it's now 7pm, he hasn't eaten since lunch so he uses the app to look for places to eat that are still open.

Michael requires an examination transcript for an internship application. He opens the app to find where the Student Records office is and when it opens in the morning so that he can request the letter before lecture hours.

3.2.2 Spotlight on design: sketching

Sketching is a tool we use with our clients to help express design ideas. It fits into a process of generating ideas and elaborating designs.

Where sketching is useful

Creation of design ideas.
Sketching is great for getting ideas down on paper so that those ideas have a tangible form rather than an abstract one.

It also helps designers to think through ideas, forcing focus on how everything fits together.

Above, an early sketch developed for UCC.

Sketching also serves as an excellent brainstorming tool where lots of ideas can be generated without worrying too much about quality. Multiple ideas are important because if you choose a design too early, there’s a good chance it’s not the right design.

Communicate & choose ideas.

Sketching also gives us an opportunity to record and share ideas so that they can be reflected upon and challenged with alternatives in response.

Ultimately, for iQ Content, sketching is a process that allows us to create design choices before we have to make one or as Bill Buxton puts it,
“Getting the right design and getting the design right”.

Benefits of sketching

Along with prototyping, this activity gives us early insights into a design before the cost of development prevents major changes being made.

Other thoughts on sketching.

Sketches should be fast to create as well as timely.

The sketch should only be as refined as the idea. A solid idea should have a more refined sketch than a vague idea, which should have a rougher, less defined look. See the examples below.

Sketches versus prototypes

A sketch is intended to be a rough design that says “this is kind of how I think it should work” whereas a prototype should be more refined with a specific description that can be validated with clients or users. That said, there a continuum between a rough general idea and a more specific blueprint, which the below diagram illustrates.
3.2.3 Spotlight on: prototyping

Prototypes can take many forms but what seems to be largely agreed upon is that it’s a good idea to get your design ideas in front of people who are similar to the target audience that you hope will eventually use your product or website for the purposes of validating and refining the idea.

The benefits of prototyping

Prototypes as a communication tool
Design is as much about a process as generating great ideas. It’s also about communicating a vision and working with a broader team to implement the vision. This means that designers have to convey the ideas in their heads to both stakeholders and developers.

Telling people about something and getting them to try it out are two very different experiences, and prototyping aims for the latter.

Prototypes can also give developers a greater sense of what's required - giving them a better feel for the feasibility as well as timelines required to deliver the design - which ultimately should lead to better budgeting and realistic timelines.
Prototypes as a design tool
The best way to test and refine ideas is to simulate the way a person will interact with the thing being designed. Prototyping helps designers evaluate ideas relatively rapidly and throw out the bad ones.

Prototypes as an early user feedback tool
Informal user feedback sessions and more formal usability tests are great ways to figure out if what you're designing -- an app or website -- has any usability problems.

Prototype forms
Prototypes come in lots of different forms.

Paper prototypes
Low-fidelity paper prototypes are great for gathering user feedback, especially if you want to let the audience know that your design is still relatively pliable (i.e. subject to change). They're also relatively quick to produce.

Prototypes can be printouts of wireframes or even a sketch. Paper prototyping is generally used early in the design process to communicate initial design ideas.

The photo on the right is a photo of a sketch. Adding multiple sketches to the photo gallery on a smartphone is a very fast and cheap way to get designs into the hands of your users.

Clickable prototypes
Clickable prototypes have an added dimension, interactivity, something that's really hard to get across on paper.
Clickable wireframes produced by tools like Axure, OmniGraffle or FluidUI (an Irish based company) give users more of a feel than just the appearance or the idea behind the app.

Prototyping interactions
Beyond a certain point, it becomes quite a burden to simulate interactivity in clickable wireframes without introducing code. For both web and mobile, we use platforms or frameworks like jQuery (and jQuery mobile) which can help build rich production-like interactivity.
In our experience, the more the polished the prototype, the better it is for getting user testing feedback or communicating the product vision to business owners.

3.3 Build.

This section includes considerations regarding technical skill sets, distribution of apps and implications to take into account for integration in existing systems and finally, makes a recommendation on the technical approach to follow in the build of a mobile app.

3.3.1 Native apps

A debate over native apps versus the mobile web has been raging for a while now with strong opinions on both sides. First, let’s focus on native apps because they’re what most users associate with the word “app”.

Above, the staff directory feature of the UCC app which we prototyped using jQuery.
A native app is coded in a platform-specific programming language (for example Objective C for iOS and Java for Android). As such, once developed, they are tied to a specific mobile platform. If you want a native app on multiple platforms you’ll need to duplicate a lot of it using the appropriate programming language.

And it may not just be a matter of coding; there are also additional design considerations. For example, Windows Phone’s Metro user interface (UI) is markedly different to that of iPhone or Android.

“People gravitate towards the products with the best overall user experience. It’s very hard for something developed in a cross-platform, lowest-common-denominator technology to provide as nice an experience as a similar native app.”

Phil Libin, CEO of Evernote

Throw in another variable, fragmentation, particularly for Android, and native seems an increasingly unattractive proposition.

Fragmentation refers to the myriad of different “flavours” and versions of Android which need to be design, developed and tested for. And in the example above where Path is not available on a particular version of Android (on one of Ireland’s most popular Androids) you can see that this fragmentation issue is anything but trivial.
What native apps offer are generally faster, more polished user experiences than is possible on web apps (though this will change rapidly). Native apps also have better access to native device features like GPS, camera, calendar, accelerometer, address book, notifications and the music player. In addition, native apps are available through the phone manufacturer’s app portal (AppStore for iOS, Google Play), offering a revenue channel as well as a distribution channel.

And the distribution channel shouldn’t be overlooked. It fits into expectations of user’s current “app culture” where they expect to find an app in their App Store rather than through the browser.

3.3.2 Another approach to apps: buy it in, don’t build it

Another opportunity within the app space is to buy in a mobile platform. Blackboard Mobile Platform is one such example which is used by Stanford in the US and UCD in Ireland.

(See our mini case study buying in a platform in Appendix II).

3.3.3 Mobile web apps

Mobile web apps are built using existing web technology - HTML, CSS & JavaScript and will rely on a server-side language (i.e. Java, PHP, Rails, Python). They are accessed through the phone’s web browser, meaning the app should work across all devices, ensuring the greatest possible reach and cross-platform compatibility.

The same base code can be used to support all devices, including iPhone and Android. However, web apps do not (yet) make use of most of the hardware features on a smartphone, such as the camera, address book etc.

In addition, web apps are not available through the various marketplaces, but through the browser.

Some of the key benefits of a web app:

1. There’s no need to install a new version of the app when changes are made centrally.
2. A common code base across all platforms allows for reduced costs of development.
3. Web-based skills are easier to access than specialised platform-specific languages.
4. Costs are less than building natively for different platforms.
5. Web apps can open links - native apps can't.
### 3.3.4 Native versus web app: side-by-side comparison

<table>
<thead>
<tr>
<th></th>
<th>Native</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User experience</strong></td>
<td>Often more polished, faster, more responsive.</td>
<td>Opens up links from email, whereas native cannot</td>
</tr>
<tr>
<td><strong>Reach</strong></td>
<td>Limited to users of the platform you develop for</td>
<td>Works on most smartphone browsers and can be made work for all.</td>
</tr>
<tr>
<td><strong>Phone capabilities</strong></td>
<td>Can access the full suite of available capabilities</td>
<td>Device orientation, location</td>
</tr>
<tr>
<td><strong>integration</strong></td>
<td>Each mobile platform (e.g. iOS, Android) requires its own language - often different developers or companies</td>
<td>Relies on browser web technologies and a server-side language</td>
</tr>
<tr>
<td><strong>Development</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Discoverability &amp;</strong></td>
<td>Available for download via App Stores.</td>
<td>Available through the web browser, can be linked directly to from email, web and SMS.</td>
</tr>
<tr>
<td><strong>distribution</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Support &amp;</strong></td>
<td>Users may not download latest version so you need to support multiple versions and make sure you communicate the “why” behind new versions - bug fixes or feature enhancements.</td>
<td>Users always have the latest version of the web app - only one version to support. However, not all smartphone browsers are equal.</td>
</tr>
<tr>
<td><strong>Maintenance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Costs &amp;</strong></td>
<td>More expensive and rely on a submission &amp; approval process</td>
<td>Less costly to develop and maintain.</td>
</tr>
<tr>
<td><strong>time-to-market</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.3.5 Approaches to mobile web: responsive design

Responsive design is an approach that has been increasingly popular over the last two to three years. The great thing about responsive design is that the techniques are easily adopted by most web designers.

The core tenet is that with one common HTML source, you can adapt the same content for mobile, desktop and tablets based on their screen-size. A very attractive proposition, but what it means is that you don’t have a separate mobile experience to that of the desktop.

**Pros:** It’s a good approach for a content-rich site

**Cons:** you can’t get much of a context-specific experience. Works only on modern smartphones.

**Where to use it?** If you’re redeveloping a website from scratch and want it to render well on mobile, desktop and tablet. Can be difficult to retrofit.

See it in action: Visit, Notre Dame’s site, [http://nd.edu/](http://nd.edu/) on a desktop and minimise the browser window to see how the design alters to fit the viewport.

3.3.6 Approaches to mobile web: server-side adaptation

Server-side adaptation is an approach which relies on device detection at the server side.
Using a device database (like Device Atlas or WURFL), the device’s capabilities can be matched to what is then sent down back to the device from the server. An iPhone may get one version of HTML, CSS & JavaScript while an older Nokia “feature phone” would get a different set. And it’s not limited to mobile, it could be desktop, tablet, Smart TV.

**Pros:** a device-specific experience can be delivered with no client-side delays. According to dotMobi, it is the most common approach to adapting content to different devices and used by Google, Facebook, Amazon, YouTube, ebay and Yahoo and more than 80% of the Alexa top 100 websites.

**Cons:** to keep up to date with the list of devices to match there are various ongoing updates (and fees) to manage. In addition, user agents can masquerade as others, to ensure that code written specifically for one browser is rendered on theirs.

### 3.3.7 Hybrid: best of both worlds?

Native versus web isn’t limited to one or the other. There’s a third option, promising the best of both worlds. Hybrid apps are like native apps in that they are downloaded and run on the phone, but like web apps, they’re written with web technologies (HTML, CSS and JavaScript).

Hybrid apps use a web view control present the HTML and JavaScript files in full-screen, using the native browser rendering engine, WebKit (but not the browser itself).

What makes hybrid apps attractive is access to the smartphone’s hardware. Access is made possible by the implementation of an abstraction layer that exposes the device capabilities via native APIs. This abstraction layer is usually via a JavaScript API and the best example is PhoneGap.

The native wrapper is compiled around the HTML, CSS and JavaScript resources, and an “interop” layer allows the JavaScript APIs to talk to the platform-specific hardware APIs.

### 3.3.8 Our recommendation: go web first, native second.

Our recommendation is to use the mobile web first, unless there is strong rationale for a native app.

1. **Greater reach & inclusiveness:** don’t restrict by the latest and greatest handsets and platforms
2. **Links:** mobile web is findable through search, email, links and SMS. Links can’t open apps.

---

3. **HTML5**: capabilities and support are being added rapidly, blurring the lines between native and web.

4. **In-house skills**: Third level institutions are more likely to have highly developed web skills within their internal teams than the more niche Objective C and Java programming languages.

5. **Cross-channel integration**: as we can see from the device ownership data in our recent survey, mobile is just one of a number of devices owned and used by students. By adopting a web approach, you’re making it easier to offer the same service on a number of different device classes, not just mobile.

6. **Lower costs**: build once, support a single version.

### 3.3.9 Skill set considerations

Organisations are more likely to have in-house web skills than specialist platform coding skills which may have a bearing on your technology choice.

It may make more sense to stick with web technologies when you have access to inhouse web development expertise. And where there’s a need for native smartphone capabilities you can consider a hybrid app rather than native languages to add this support via supporting APIs.

In terms of maintenance or support as well as the addition of improvements or fixes, these also need to be factored into to the total effort of managing an app.

### 3.3.10 Distributing apps via app stores

Each of the different platforms have their own App Stores where native or hybrid apps are distributed. The following are the probably the most common:

1. Apple AppStore
2. Google Play
3. Windows Phone Marketplace
4. Blackberry App World
5. Nokia’s OviStore

To get into the app stores for the different platforms, the process looks like the following:

1. Register as a publisher
2. Submit the app for approval
3. Deploy and support
3.3.11 Register as a publisher for each app store

For most of the stores, you'll need to first register as a publisher. The process for Apple and Android are somewhat similar, though the turnaround times can be anything from days to weeks.

There are also one-off costs associated with registering as a publisher:

- Apple App Store: US$99
- Android: US$25
- Windows Phone: US$99
- Blackberry App World: US$0
- Nokia’s OviStore: €1

From there, there are no additional costs and you can then distribute your app from the various app stores. However, if you charge for your app, the app stores can charge a percentage of the app price (Apple charges 30% for example).

3.3.12 Build with analytics

Just as you would with a website, you should carefully consider what aspects of user behaviour you should measure within your app.

Like web analytics, business objectives and KPIs should be the drivers for what you measure, why you measure it and what you do with the insights.

Some of the metrics most commonly measured within apps are:

1. Number of downloads
2. Ratings
3. Screen flow, user interaction within app
4. Where users are (location)
5. When users use (i.e. when they use the app)
6. How often they use
7. How long they stick with it
8. Custom events (this could literally be anything you define and is very specific to the app, like sharing a piece of content via social media, “checking-in” etc.)

What tool to use for app analytics?

Although Google Analytics and other traditional web analytics providers come with an SDK for tracking app data, app behaviour can markedly differ from the web behaviour.

Some examples like dynamic content or screen flows, how users engage with mobile apps versus websites mean that the web style reporting of “pageviews” that are the basis of web analytics aren’t really relevant to apps.
Apps also have some very specific analytics needs. The analytics data should:

1. **Cache when offline** (to be sent to host when back-up)
2. Be **cross platform** to allow for similar metrics to be taken in the same way from different phone platforms
3. Allow **segmentation** of users (e.g. you can view data by those who are using for first time or those who have it a month)
4. Accommodate **advertising APIs**: to allow for integration and tracking of any mobile ad platforms
5. Be **easily implemented**: development phased needs to focus on the building the core features of the app, not the analytics.

With the above considerations in mind, we’d recommend more app-specific analytics packages like Localytics or Flurry.

**3.3.13 Submit app for approval**

When you’ve built your app, you’ll need to submit it to the AppStore for approval and there’s no guarantee it will be accepted nor how long it will take to be accepted.

A university app which includes Students’ Union bar promotions could be refused submission because you got the age rating of the app wrong.

Another example of app refusal is if an app simply emulates a mobile browser by serving web pages (with no real “on-device” features).

The reasons could be manifold, so the key point is to allow for this contingency.

**3.3.14 Discoverability within the app store**

Once your app has been accepted to the various app stores, it needs to be made “discoverable”. This includes in-app store promotion, using the most relevant copy and high quality images while also working with your various web properties to drive users to download the app.

For apps in a competitive niche, there’s even a new equivalent to the search engine optimisation industry springing up called “App Store Optimization”. One provider, whom we haven’t used, promises to “analyze and optimize your app store presence and beat your competition”.

3.3.15 After launch

After launching the app, keep an eye on both the ratings/reviews for any signs of negative experiences brought about by bugs or crashes as well as any other feedback. These are calls to action to make any requisite fixes and improvements to your app.

It’s also worth planning your app updates. Remember, any update is a new update for your users, so make sure they’re critical and not too frequent. And, finally, make sure you let users know the reason for the update whether it’s critical bug fixes or enhanced features.

3.3.16 Systems integration

Regardless of any app/website there will be a dependency on multiple and disparate systems from which to source content or data.

This is arguably one of the toughest challenges. However, sourcing data from various databases for a mobile-specific project means that data should be available for other uses.

This is very much along the lines of the principles of “Open Data” - opening up data in systems so that it can be interrogated and used in lots of different ways.

Solution to systems integration: Edugate

Integrating with campus authentication systems may be simplified by utilising the common authentication method provided by Edugate, this could provide the mobile application or mobile web app with personalised user data from the campus directory which could be used when querying other campus systems for further personalised data (e.g. a student’s timetable). See the UCC case study for further details.

3.3.17 Overall effort required

This very much depends on what’s being designed and built, the platform choice (web or native) and the degree of systems integration required.

It also depends on project approach, how and where you partition the work, if you use an agile, lean or a waterfall approach.

For the purposes of the UCC prototype in Section 5, the following is a breakdown of the time taken:
Discovery & analysis: 3 weeks  
Design: 3 weeks  
Prototype development: 3 weeks

Full development would take longer, particularly integration of staff directory, locations and student timetables. There would also be a good deal of work managing and co-ordinating the content production process.

4.1 Promoting the app

Within a campus setting, there’s a relatively “captive” audience, which makes it easier and cheaper to communicate the launch of a mobile app. That said, it’s worth working together with Communications/Marketing to develop a plan for launch.

4.1.1 Soft launch

The first thing to think about is a soft launch - that is going live and letting the app settle in before publicising it to all and sundry. This gives you the opportunity to address any bugs or other issues and test with a segment of users.

In the case of native apps, it also gives you breathing space to allow for App Store approval which can take anything from 3-6 weeks.

4.1.2 Email announcements

Assuming you have a mobile number or email address for everyone on campus, you can communicate with them via SMS or email. A large proportion will be using their smartphones to check email - so there’s a clear call-to-action to visit the mobile web app or download it from the app store.

4.1.3 Traditional offline promotion

This could include posters in cafeterias and other locations around campus, induction packs for new students,

4.1.4 On the existing website & social media

To communicate to both students non-students and non-staff (potential students or visitors to campus) you'll want to draw attention to the app on the existing website.

Another online channel to promote it is via the various social media channels.
4.1.4.1 Interrupting mobile visits

If they're visiting the main UCC site on a mobile device, you can also interrupt their visit (with caution) to draw attention to the presence of the app.

With the example on the right, from The Irish Independent, when visiting from a mobile, they give the user the choice between an app, a mobile site and the full site.

A word of caution, however, in the case of installed apps this interruption can't know if they already have it on their phone so it could be frustrating rather than helpful.

4.2 Measurement

After launch the main metric or KPI will most likely be “adoption”. This metric means a few different things. Within the App Store landscape it will mean downloads initially, but that’s only part of the story - an adoption KPI should also include usage i.e. what’s the ratio of downloads to being used. Other metrics are what areas of the app are being used and not used as well. Mobile app analytics tools like Flurry can also report on app crashes to help you proactively manage any bugs.

Where there’s no concept of a “download” i.e. it’s a mobile web app then we can revert to standard web analytics metrics like visits, new versus returning visitors, top content reports etc. This data will be available within tools like Google Analytics, once you’ve configured it correctly.

4.3 App store ratings & reviews

For apps downloadable from the various app stores, ratings and reviews give an indication of how well the app is received. Within the written reviews there may be some insight as to usability issues or bugs for the app. It’s worth monitoring the reviews on a regular basis to find areas you can focus on with qualitative user testing, user interviews and surveys.

4.4 User feedback

Complimenting analytics and reviews & ratings, we think it's far more valuable to get feedback directly from your users and in a number of ways - surveys for larger samples
and user testing to get some deeper insight. Any insights or requests can then be evaluated to see if they’re suitable for inclusion into a development roadmap.

4.5 Managing a roadmap

From both regular surveys (every 3-6 months) you can determine where the app meets user needs and also where they may be gaps where it doesn’t quite fulfill their needs.

You may also discover areas of the app which are under-utilised. In this scenario user testing can help you figure out why and address the issue by adding it to your roadmap for development.

4.6 New devices, operating systems

The only prediction we can make on technology is that it will continue to change at pace. From new operating system versions to new devices, each bring their own challenges to both native and web apps, particularly so for native. What it means is that you should have a programme of regular testing on new devices and operating systems and where they’ve forced changes in design and development, regression testing on older operating systems and devices.

For apps, one way to stay proactive is to take part in their respective developer programmes where you can test on preview releases of new operating system versions.

4.7 Content & data governance

In our experience many organisations face a common problem around content governance - content isn’t always being updated by those responsible for it. The impact of content going out of date is that the credibility of the app can then begin to suffer.

Overall: ownership & governance

Regardless of mobile app type (native or web app) there is a clear need for governance so that each of the above activities can be planned and managed in an orderly way.

A good way to think about governance is breaking it into four levers:

1. Skills & staffing: what skills are required to maintain & evolve
2. Team structures: what roles are assigned and how is the team structured and what leadership system needs to be in place to make sure the app can be maintained to a high standard.
3. Tools & technology: what platforms are supported and which are not.
4. Processes: the systems and documentation for carrying out the activities of governance

The activities in managing an app are similar to those for managing a website.

1. Planning, creating & publishing new content.
2. Designing, coding & testing new features.
3. Day-to-day supervision, QA, feedback, performance monitoring, etc.
4. Infrastructure: management of the hosting, security, software updates etc.

Each of these levers and activities need to be tailored to suit your own governance system.

The team could be structured along the same lines as a web team, and in practice it’s often the same team managing both. Whether the roles sit within or external to the organisation depends on resources and staffing.
Above, a well resourced web team which also relies on external contractors where necessary.
<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designer</td>
<td>There are two main roles wrapped up under the term. A User Experience (UX) or Interaction Designer is responsible for designing flows and interactions on the app or website, should also be able to facilitate user research like user testing, evaluations, surveys while a Visual Designer is responsible for the look and feel.</td>
</tr>
<tr>
<td>Content producer</td>
<td>The content producer sources content from various business owners and other sources, existing content, images, PDFs, video. They then produce clear, concise content which reflects what users need in a style suited to the medium.</td>
</tr>
<tr>
<td>Developer</td>
<td>This role could cover a range of activities (or may need to be split by speciality) from HTML, CSS &amp; JavaScript to native language coding to database programming and systems integration.</td>
</tr>
<tr>
<td>Editor</td>
<td>We've used the term editor to reflect the role that is required to run and manage the web team - ultimately, they're a project manager, a product manager and an editor rolled into one and oversee the production of content, design and development.</td>
</tr>
</tbody>
</table>
5. UCC case study

The key contributor from UCC was Barry Foley, Systems Administrator with Student IT. For the full length of the project Barry led the efforts from within UCC.

In particular:
- deploying and reporting on student and staff surveys
- coordinating and participating in the UCC stakeholder interviews and design workshops
- setting up the technical side of hosting the prototype

From an iQ Content perspective, the following was the effort required to research and develop the prototype.

- **Discovery & analysis**: 3 weeks
- **Design**: 3 weeks
- **Prototype development**: 3 weeks

The approach taken for the UCC case study is that of the user centred design approach as discussed in Section 3.

The goal for UCC was to get as far as a demonstrable prototype to act as a catalyst within the organisation to prioritise a mobile strategy. For that reason, we took the prototype as far as a high fidelity working prototype.

The case study covers the initial discovery and analysis phase, the design phase and finally the prototype build phase.

---

5.1 Summary of our work with UCC.

**Students own multiple devices**
The main ones being mobile and laptop (99% and 93% respectively).

Smartphone penetration in the student population is high at 79% (in UCC, see Appendix I). This compares with 43% for the adult population, from a Google study.

**The mobile landscape is fragmented**
What’s clear from the device branding is that there is a spread of different platforms - iOS, Android, Blackberry, Symbian etc. This will become a critically important factor when it comes to choosing the technology on which an app should be built and supported.
Mobile use cases

From the synthesis of the discovery activities we did with UCC (the stakeholder interviews, staff & student surveys and peer analysis), we were able to determine a good view of the “desirability” of the design candidates for the mobile app (see Appendix III for all app candidates we identified).

Design candidates

Below, is the initial top candidates, starting with the most “desirable”:

1. Timetables
2. Campus map/directory
3. Exam results
4. Course notes (Blackboard)
5. News & events
6. An app for alumni
7. Mobile version of web portal
8. Club & society information
9. Food menus/restaurant information
10. Car parking
11. Assignment Submission Reminders
12. Library app
13. Student voting
14. Grant updates

Many of these app ideas are inline with what we saw in our peer analysis. See the full description of each of these app ideas in Appendix II.

App decision made: campus map

The ultimate decision we made for UCC’s prototype was to focus around the campus map.

An important point is that a campus map is not just about way-finding in an unfamiliar environment, i.e. getting from one place to another. Presumably, students become familiar with their environment so would a map become less relevant?

The following quote from an internal UCC stakeholder illustrates how a map based app is about more than way-finding.

“Mapping is needed. It’s not just A to B - it's what services are available using visual communication”

A campus map goes beyond way-finding and offers:
- **Greater reach**: it’s not confined to the new undergraduate student, it will also support staff, prospective students, alumni and campus visitors.
- **Greater utility**: more “useful” than news or events for finding location and time-specific information.
- **Greater scale**: can scale to allow for greater layers of information both within and beyond the campus, from shops, events, parking spaces, PC availability etc.
- **More opportunities for collaboration & funding**: as a result of the greater potential scale, there are opportunities to work with various organisations both internally and externally and who may be willing to part-fund the development of the app.

### 5.2 Discovery & analysis phase.

The discovery phase consisted of the following in order of activity:

1. Stakeholder interviews
2. Student survey (aimed at all students, but with a high undergraduate response)
3. Staff survey (a mix of full and part-time staff in both administration and academia)
4. Peer analysis
5. A prioritisation of mobile app candidates

#### 5.2.1 Stakeholder & user interviews

The objective for both sets of interviews was to get a deeper understanding of what the app should be and in particular:

- Identify the core user group for a mobile app.
- Understand any internal objectives and requirements for a mobile app.
- Understand how various groups currently feel about the challenges in creating a mobile app.

**Who we interviewed**

We interviewed 10 people over two days, with a mix of academic, administrative and service staff. In addition, we also interviewed three students (two current and one recently qualified).

**The themes we explored.**

We weren’t overly prescriptive in having a fixed set of questions to ask because this can actually limit insight: you’d only get answers to the questions you ask.
Instead, we focused on key themes:

- **Who**: Who will be the ultimate mobile users that we should focus on?
- **What**: What should we design for the users we’ve identified?
- **How**: How should a mobile app be implemented and what are the technical constraints?
- **Why**: In what way would a mobile app serve the business objectives?

**Stakeholder & user interview outcomes**

1. **The target user**

   “Focus should be on undergrad because of the sheer number. We normally weight things towards their needs”.

   While this opinion was shared amongst most of the stakeholders there was also broad acknowledgement that a campus map or directory would have broader appeal.

2. **The opportunities & challenges**

   One of more feasible of the app ideas was a **campus map**.

   Initially, we challenged a stakeholder about this “diminishing usefulness” of a campus map (i.e. that students and staff get to know the place), one stakeholder held a very interesting and valid view,

   “**Mapping is needed. It’s not just A to B - it’s what services are available using visual communication**”.

   Another staff stakeholder added,

   “**University experience doesn’t stop at the campus boundary, it’s also about Cork. You have international students, Dublin, Kerry students, so mapping can extend beyond campus to restaurants, pubs etc.”**

   Another was able to pinpoint some of the perennial problems with being on campus, synonyms for buildings which come up in timetables and availability of services.

   This frustration is evident from the current website, where complex internal coding and jargon leaks out to unsuspecting users. The following is taken from the UCC website:
One stakeholder summed it up best.

“Building codes are daft”.

Another elaborated a bit more,

“What do the building names or codes mean? The Kane versus the Science building, and what’s in their building i.e. labs with the newest machines that are idle, or a restaurant in the basement that no one knows about.”

### 3. The technical considerations

“We have to concentrate on getting fundamentals in order - must get the data in order first before we do anything, not having silos of information in different places. Otherwise we inherit the problems of desktop”.

Another feeling among stakeholders, many of whom were technical service staff, was that the app should be built on open web standards (HTML5, JavaScript and CSS) rather than for proprietary “native” platforms.

There’s a mix of ideology and practicality here, which we’ll discuss later in the technology section.

#### 5.2.2 Online surveys

The purpose of the UCC student survey was two-fold:

1. to get some quantitative data (“hard data”) on who are respondents were and what types of devices they use.
2. to get an understanding of the students’ needs (more qualitative than quantitative in nature).

**Student survey**

**What we asked: the quantitative questions**

1. Age range.
2. Course Type.
3. Which of these devices do you own?
4. What type of mobile phone do you use?
5. What brand of phone do you use?

These questions were closed (user must select from a range of answers) and give us some hard data about our users’ context - who they are and what devices they use.

**What we asked: qualitative**

The next set of questions were also “closed”, but more qualitative in nature. We wanted to get a feel for students’ perspective on the “desirability” for a set of mobile apps which we had previously identified at the outset of the project through meetings with HEAnet and UCC stakeholders.

We asked them to rate (1=least useful, 5=most useful) some app ideas that they thought would be the most useful for them.

**What we asked: open verbatim question**

The final question we asked was an open “verbatim” question to get a sense for how students felt, but more importantly, in their own words. This piece is critical for a number of reasons - you can gain a deeper insight because respondents are more expressive in an open question - this can lead to greater empathy - i.e. really understanding the needs and frustrations of those we’re designing for.

**Q: Please list the services the University should offer through a Mobile App.**

Using Wordle, we visualised the word frequencies from the verbatim answers by creating a word cloud. This communicates, in a quick glance, the most frequent terms used by students.
Other student survey findings

From the survey, we found that 65% of the respondents were between 17 and 22 years old, with a further 19% under 25. Overall, 85% of respondents were undergraduates.

Students own multiple devices, the main ones being mobile and laptop (99% and 93% respectively).

We can also infer that smartphone penetration in UCC’s student population is 79%. Interestingly, for comparison purposes, a recent study commissioned by Google puts smartphone penetration in Ireland at 43% while another study puts the adult smartphone penetration at 59%.10

In terms of what phone, what’s clear is that there is a wide spread of different platforms, the most popular being being Android (44%), iOS (26%), Blackberry (11%) and Nokia (15%).

Staff survey

The staff survey contained open questions because as “interviewers”, we wanted to get at the kernel of what was most important but also what’s most difficult from those who are most familiar from their day-to-day experiences on campus.

In addition, because the staff are so diverse, it gave us the opportunity to get ideas from different perspectives.

The four open questions we asked were:

10 Our Mobile Planet; Ireland, understanding the mobile consumer, May 2012
1. Who should we design for?
2. What's the first thing you would do and why?
3. What's the second thing you would do?
4. Any other thoughts on mobile?

Staff survey outcomes

Staff were mainly a mix between academic (ca. 48%) and administrative (31%) with research and services also included.

In addition to app ideas, many staff were able to clearly articulate how it should be done as well as some of the internal challenges or obstacles within UCC.

Below, two examples.

“Please optimise for multiple operating systems (i.e., at least Android and iOS), ideally in a form that would render well both on smartphones and tablets”.

“The problems encountered when developing the portal with regards to single sign on and so forth would be the same as those encountered when developing a mobile app. We need underlying systems that can talk to each other before we should spend our time developing an app”.”

A staff survey response.

“We’ll cover it later but what comes out loud and clear is that solving the content source or systems integration problem is the strategic challenge for any organisation when developing a campus-wide, multi-service app.

The following is a visualisation of what staff typed in as ideas.
5.2.3. Peer analysis - benchmarking similar apps

We analysed four universities’ mobile offerings to identify what they’re doing well and what we can learn from them.

We cherry-picked four mobile apps that we thought would give a good representation:

1. Stanford
2. UCD
3. Harvard
4. m.bristol.ac.uk (a mobile website in beta, not an app)

A common dashboard approach

The first three from the list above took a similar approach - a dashboard design which contains a series of discrete features or “applets” focused around everything from academic support to the opening hours and menus for campus eateries.
They provide literally “everything under the one roof” with room to add more when it becomes available.

**Note:** Stanford and UCD use Blackboard which has an extendable SDK, via which you can add more apps. Read more on UCD’s experience with Blackboard’s mobile platform in Appendix II.

Blackboard Mobile Platform is a proprietary system, which can be licensed on a campus level. There’s also another model where students are charged to download it from the Appstore, reducing the development costs of the app by deferring it to the students. The cost is around $2 per student per year, though this may depend on campus size.

The apps are also very broad in their appeal, not focusing on one specific user type nor one specific “use case”.

The following are the most common features among them.

1. Staff directory
2. Map
3. Library
4. Where to eat
5. Educational content
6. News and events
7. Course information

**Going beyond native apps**

What’s also worth noting about Stanford and Harvard is that they’re not limiting themselves to just their platform specific native apps - they both have mobile websites in addition to their apps.

In Stanford’s case their mobile website differs from their app quite considerably.
Above m.stanford.edu and m.harvard.edu, websites designed for mobile browsers.

Bristol (m.bristol.ac.uk)

Bristol University’s efforts in the mobile space warrant special mention as their project has been openly and well documented\textsuperscript{11}.

Other points to note:

**Web approach**

\textsuperscript{11} Introducing the MyMobileBristol Project: http://mymobilebristol.ilrt.bris.ac.uk/2010/07/13/introducing-the-mymobilebristol-project/
MyMobileBristol write-up: http://www.ariadne.ac.uk/issue67/jones-et-al
User feedback: https://mmb.ilrt.bris.ac.uk/display/mmb/User+Feedback
They have taken a web approach - not limiting the services they provide to specific platforms. This is an approach that was favoured by internal UCC stakeholders.

**Multiple, disparate sources of data**
They identified open data as one of the key dependencies of a useful app and built the infrastructure which can cater for multiple, disparate sources of data.

*Above, schematic of Bristol's mobile platform and how it deals with disparate data sources.*

**Beyond the campus**
They acknowledged their place in the environs, not limiting their scope to the confines of the campus but of the greater Brighton area. This approach also made for greater collaboration with Bristol City Council.

This aligns with our stakeholder interviews in UCC, where one user commented,

> “The university experience doesn’t stop at the campus boundary, it’s also about Cork. You have international students, Dublin, Kerry students, so mapping can extend beyond campus”

**Bristol's open source approach**
They've released their platform as open-source so there's opportunity for wider adoption and reuse by other organisations.

**Bristol's user feedback**
What comes out strongly from user feedback is the desirability for blackboard and timetables to be included, that is “closed data” in addition to the open data.
What’s also noteworthy is users’ expectations that the app should be available for download in respective App Stores rather than through the browser\textsuperscript{12}.

5.2.4. Discovery & analysis wrap-up

Summary of this section

In this section we conclude on:

1. Who we should design for
2. What we should design

1. Who we should design for

Through the stakeholder interviews and staff survey, the main user groups mentioned throughout were:

1. Students
2. Prospective students
3. Staff
4. Visitors to campus
5. Alumni
6. Collaborators (e.g. other universities, research institutes, companies) that UCC works closely with.

From these user groups, the key design target coming out of the stakeholder interviews was the undergraduate student.

However, as we’ll see later with the mobile app candidates, many of them would serve more than just the undergraduate student.

What we know about our users

Device ownership
Students own multiple devices, the main ones being mobile and laptop (99% and 93% respectively).

\textsuperscript{12} \url{https://mmb.ilrt.bris.ac.uk/display/mmb/User+Feedback}
Smartphone penetration
Smartphone penetration in the student population is 79% (an anecdotal comment from one member of staff was that it was around 20%).

Device platforms
What’s clear from the device branding is that there is a spread of different platforms - iOS, Android, Blackberry, Symbian etc. This will become a critically important factor when it comes to choosing the technology on which an app should be built and supported.

2. What we should design

Now that we’ve a clear audience we need to look at how we can meet their needs.

From the synthesis of the discovery activities (the stakeholder interviews, staff & student surveys and peer analysis ), we have a good view of the “desirability” of the design candidates for the mobile app.

Design candidates

Below, is the initial top 14 candidates, starting with the most “desirable” - from both surveys (those mentioned most) and the staff & student interviews (where options were discussed in depth).

1. Timetables
2. Campus map/directory
3. Exam results
4. Course notes (Blackboard)
5. News & events
6. An app for alumni
7. Mobile version of web portal
8. Club & society information
9. Food menus/restaurant information
10. Car parking
11. Assignment Submission Reminders
12. Library app
13. Student voting
14. Grant updates

In addition, many of which are inline with the features within the apps we saw in our peer analysis. See the full description of each in the Appendix III.
Moving from a long list to a short list

With some strong candidates identified and some consensus between stakeholders, staff and students, the question is where we should focus our initial efforts? We could do all of these things, of course, but probably not all at the same time. We need to start with the most important, those “anchor tenants” or “killer apps”.

So we need to reduce this list and prioritise what we must do now versus we could do later and, finally, what we’re not going to do.

How do we prioritise?

While we have now have a good understanding of the most desirable ideas, we need to introduce another variable, feasibility, so that we know the idea can actually be implemented and supported.

Above, the best ideas should be both desirable and feasible.

Step 1: List the app opportunities or features.
In a spreadsheet, a list of the main opportunities to be evaluated.
Step 2. Score those ideas for desirability and feasibility
We produced a scoresheet to help prioritise ideas.

<table>
<thead>
<tr>
<th>Design Prioritisation worksheet</th>
</tr>
</thead>
<tbody>
<tr>
<td>App idea or feature</td>
</tr>
<tr>
<td>Timetable</td>
</tr>
<tr>
<td>Car parking</td>
</tr>
<tr>
<td>Results</td>
</tr>
<tr>
<td>Exam Timetable</td>
</tr>
<tr>
<td>Library (Bookable)</td>
</tr>
<tr>
<td>News &amp; Events</td>
</tr>
<tr>
<td>Campus Map</td>
</tr>
<tr>
<td>Directory / Contacts</td>
</tr>
<tr>
<td>Clubs / Societies</td>
</tr>
<tr>
<td>Deals / Discounts</td>
</tr>
<tr>
<td>Catering</td>
</tr>
<tr>
<td>Restaurants / Bar (Menus / Offers etc)</td>
</tr>
<tr>
<td>Electronic Voting (SU)</td>
</tr>
</tbody>
</table>

The scoresheet is based on both desirability and feasibility.

- **Desirability**: Desirability is about what people want or need most. We’ve split this further between for users and desirability for the business, because without the latter, there is no “business case”.

- **Feasibility**: Feasibility answers the question around “can it be done?”. Again, we’ve split feasibility, looking at it from two perspectives - technical and resources. Technical is around systems, new or existing, development, integration issues and budget. Resources is more about ownership and roles.

With this approach, we can evaluate an idea like “student daily deals” - which could have a high desirability for users (particularly students), but could yield no real benefit to the business.

Using another example, a “food menu” could be really desirable for students and indeed, the business. And while it could also be really easily to implement, there may be no one to manage the content updates regularly enough for the app to be useful.

**How to use the scoresheet**

1. Populate with all the top ideas or features that were previously generated.
2. Each stakeholder has a copy of the sheet with the ideas listed to score themselves.
3. Stakeholders score the ideas for both desirability & feasibility.

There’s a point available for each idea on the list which they are free to distribute as they see fit.

So, if there are 20 app candidates, you can spend all your 20 points on just one idea, marking it clearly the most desirable or feasible. Or, you could spread 20 points amongst 4 apps, giving them 5 points each but then they’d be as important as each other, with none
standing out as the priority. However, in our experience, one of the other columns normally tips the balance.

“What if I can’t score an idea?”

If desirability for the business can’t be easily scored then this could be a finding on its own - that we’ve the wrong people in the room and that we need to identify true business owners or “sponsors” and involve them in the process (they’ll need to be involved anyway or we’re making decisions in a vacuum).

Keeping it simple.

The principle is to keep scoring simple so that decisions can be made relatively quickly while not getting too bogged down in the detail. Ultimately, this process must force us to choose.

The “detail” can still be considered, but more to justify the scoring. For example, the scores may come from any knowledge staff have about constraints like poor systems integration or not having enough staff.

Step 3. Add up the scores and graph the results

What we should get out of this are candidates we should focus on right now (most desirable and most feasible), candidates we may get to later and candidates we shouldn’t even consider.

Within UCC, the exercise was run with seven stakeholders scoring on their own. Scores were then aggregated and we produced a graph plotting desirability (y-axis) versus feasibility (x-axis) to visualise the competing priorities. We divided up the graph into four quadrants:

1. Do now
2. Do next
3. Come back to it
4. Don’t do
5.2.5 App decision made: campus map

The ultimate decision we made for the design candidate to focus on for our prototype was the campus map.

Why?

**Greater reach**: beyond the undergrad student, it will support staff, prospective students, alumni and campus visitors

**Greater utility**: more “useful” than news or events for finding location and time-specific information.

**Greater scale**: can scale to allow for greater layers of information both within and beyond the campus.

**More opportunities for collaboration & funding**: as a result of the greater potential scale, there are opportunities to work with various organisations internally and externally and who may be willing to part-fund the development of the app.

5.3. Design phase

Design brainstorming workshop
After discovery we should have a fairly good idea on what to build and why. The next activity was a day long collaborative workshop with designers and stakeholders who allowed us to identify the things that would be compelling within a specific design’s context.

**Step 1: Generate ideas**
Get everyone in the room to work on their own, writing all the ideas down on post-its (one idea per post-it).

**Step 2: Create an idea wall**
One-by-one get everyone to tick their post-its up on the wall.

![Image of post-its on a wall]

**Step 3: Group the ideas**
Once everyone has put their post-its on the wall, the next part of the exercise is to group them.

The task is quite simple, on a blank piece of the wall, everyone takes a turn to review the other participants’ post-its on the wall and group any post-its which they think are related.
Up to this point, there still has been little or no “group discussion”, just individuals in the same room working alone.

**Step 4: Label the groups**
The next task is to give each group a name and, again, everyone in the room does this on their own. We'll end up with ideas grouped together with multiple labels.
**Step 5: Scoring the groups**
We then move on to scoring the ideas - simply asking each workshop participant to pick their top three groups of ideas and mark them accordingly.

At this point, we haven’t allowed “groupthink” to set in nor have we allowed the loudest or strongest opinion to dominate proceedings.

The benefits of this approach is that, without any drawn out discussion, you can begin to identify what the most important things are i.e. the things that the design should focus on first.

It’s only then that the interesting conversations can happen, because we’re only discussing what’s most important and not wasting time on anything that isn’t.

An additional benefit is that it highlights how everyone can participate as a “designer”. With this technique, the designer plays the role of facilitator - the point being anyone can (and should) be able to run this kind of workshop.
5.3.1 From sketches to wireframes

Iterative design process
Design was iterative in nature, starting with sketches and finishing with a final polished design.

With an iterative process, designs are produced and evaluated quickly so modifications can be made before major investment of time and money.

Each iteration becomes more elaborate or polished as the design is refined, moving from rough sketch to polished visual design.

The benefits of this approach:

1. **Fast**: lots of designs can be explored early before settling on one
2. **Communicated**: Designs are communicated and evaluated early so there are no surprises at the end of a drawn out design period. Any issues can be caught and addressed.
3. **Collaborative**: stakeholders are involved and are part of the solution, rather than just signing off on it.
Above, an early sketch developed for UCC.
Above, a wireframe (schematic) which describes the form & behaviour of one aspect of the proposed UCC mobile app.

5.3.2 Prototype

Above, the staff directory feature of the app which we prototyped using jQuery.

In our experience, the more the polished the prototype, the better it is for getting user testing feedback or communicating the product vision to business owners.

5.3.3 Visual design

Visual design considerations

When designing for mobile there are a number of important considerations:

1. **Screen size is limited**: which means focusing on core content and actions.
2. **Bandwidth is limited**: speeds are often slower on mobile, which means file sizes of HTML, CSS, JS and images should be kept down.
3. **Make content the action**: an example in this context is allowing users to pan and zoom a map using touch gestures rather than buttons, freeing up the screen for the whole map.
4. **Make touch targets larger**: touch targets should be a minimum of 29px wide and 44px tall (iOS guidelines for iPhone).

For the UCC prototype we used a dashboard home screen with familiar icons, a common design pattern for mobile apps.

Visual design produced

The following are some of the visual designs produced.
The home screen - a common design pattern with familiar icons for each section.

Map takes up most of the screen with familiar touch gestures used for panning and zooming rather than buttons.

Use of photography to aid orientation when looking for building details.
5.4. Prototype build

Technologies chosen

We deployed the prototype to [http://mapp.ucc.ie](http://mapp.ucc.ie) using HTML5, CSS3 & JavaScript (jQuery mobile).

They were chosen for the following reasons:

1. **Faster and cheaper to maintain & deploy**: designed and developed once which means all platforms pick up the latest version via their mobile browser.
2. **Distribution**: via the web which means a greater reach to multiple platforms via the browser
3. **Shareable**: content can be more easily shared with others via hyperlinks.
4. **Skills**: web skills are more common within the university’s staff (as opposed to native coding languages).

However, some of the drawbacks of this approach are:

1. No offline access
2. Limited or no access to a smartphone’s hardware features

**Deployed to UCC with Edugate**

We successfully tested the prototype with a secure server in UCC using HEAnet’s Edugate service. The purpose here was to simulate the app as it would appear when live. Authentication would be a requirement for personalisation (such as schedule).
Benefits of Edugate

Security
Transit or traffic between user and identity provider is encrypted and Log/SP, SP web server logs do not record passwords.

Single view of directories in an institution.
If an institution has more than one directory service, Edugate aggregates them all in a single view. Interaction between application and directory is a standardized to Edugate instead of having to use different connectors for different directories (i.e. LDAP/DB/AD).

Single user respresentation.
Edugate represents a user in a single standardized manner

Drawbacks

User experience
User has to enter username/password for every session expiration. If user credentials were permanently stored in the phone this would not be the case.

Visual experience
User is redirected to identity provider where the visual design may not be consistent with the web application (though CSS can be provided)

Speed in authenticating.
There’s now a dependency between the identity provider and the web app so a failure of the identity provider would prevent access to parts of the app which reaquire authentication.
Above, how the final app could be integrated with different systems within the UCC environment.
6. Conclusions: Lessons learned

At the end of the project, there are clear lessons learned from the strategic approach through to technical implementation. We cover the main lessons below.

6.1 Mobile, an opportunity to focus on the essentials

There are inherent constraints with the smaller screen size on a mobile phone. And with a smaller screen size, you have a smaller canvas to work on, which forces focus & priority on what matters most to your users.

6.2 From Mobile app to mobile web

At the outset of the project, the focus was on a “mobile app”.

However, during the research and analysis phase we discovered a number of things which caused us to re-evaluate what a mobile app could actually mean.

On one hand the app is available through to a limited set of platforms via their respective app stores. On the other hand, if done through the web and available via mobile browsers, greater reach is possible.

However, trade-offs have to be made between better, faster experiences on native platforms which can take full advantage of all the smartphone features and greater reach via the mobile web where there is a very limited set of hardware features.

Whatever approach you take, the important piece is relevance on mobile.

6.3 User centred approach

Without speaking to users we can bring biases and assumptions which can cloud our judgment. Mapping is one such example. One bias we brought with us was that after a few weeks, the core audience of the app, students, would have learned their way around campus.

However, users we interviewed put us straight, saying that college life does not begin and end at the campus boundary.

This example highlights arguably the greatest benefit of user centred design - the empathy it engenders - walking in the shoes of your users.
6.4 Surveys

Our survey yielded a result that 79% of students had smartphones. The number had been estimated to be somewhere between 20% and 50% by stakeholders we talked you.

What this demonstrates both how quickly the mobile landscape changes as well as the need for regular data - we’d recommend a new survey every 6 months.

6.4.1 Getting greater survey response rates

Another practical aspect we discovered is how shorter surveys can yield higher response rates.

In addition, deploying a survey via Google Docs (where the institution uses Google Apps) could also lead to higher responses because the survey questions can be embedded in the email and can be responded to from within the recipient’s inbox.

6.5 Prototyping before production

Prototyping uses design as a means to explore, not just develop.

Through prototyping you can quickly discover underlying requirements (like the need for systems integration to happen before the features of the app could ever be realised).

Prototyping also gives stakeholders and users the opportunity to engage in a way they never can with functional specification documents or abstract ideas. Designs can be tested and validated earlier and iterated, improving the product beyond what would be normally possible.

6.6 Don’t underestimate the systems integration effort

Just getting at the data that a mobile app needs can be a challenge. Something as simple as a timetable can be a complex beast to tame.

It may involve systems to begin to talk teach other via new APIs or an identity management provider to be rolled out across campus before personalisation for an app is possible.

Arguably, this level of systems integration should be done anyway which will facilitate better user experiences across all channels, not just mobile.
6.7 Buy versus Build

Buying in a platform such as Blackboard Mobile platform is an attractive proposition.

The key benefit is that it allows you to focus on the systems integration effort rather than the design of the app.

Other benefits:
- predictable fixed costs
- a UI which has been tested and iterated with other clients

Drawbacks include:
- Platform lock-in
- A one-size fits all (or limited scope to customise)
Appendix I: UCC Student Survey Data

Deploying the student survey

We deployed the student survey via a Google Documents Form. In UCC's case, where they use Google Apps, this meant that the survey was literally embedded in the email the student received so they could start interacting with it straight away, without having to click off somewhere else.

Response rate

There's no such thing as a “typical” response rate because it depends on a number of variable: the audience, the incentive, the type and number of survey questions. That said, with 1,240 responses from a possible 5,250 students (24%) it’s more than enough to uncover good insights and spot any emerging patterns.

Age range

17-18  
19-21  
22-25  
26-29  
30-39  
40-49  
50-59  

<table>
<thead>
<tr>
<th>Age</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>17-18</td>
<td>7%</td>
</tr>
<tr>
<td>19-21</td>
<td>58%</td>
</tr>
<tr>
<td>22-25</td>
<td>19%</td>
</tr>
<tr>
<td>26-29</td>
<td>5%</td>
</tr>
<tr>
<td>30-39</td>
<td>5%</td>
</tr>
<tr>
<td>40-49</td>
<td>3%</td>
</tr>
<tr>
<td>50-59</td>
<td>1%</td>
</tr>
</tbody>
</table>

Course type

Undergraduate
Postgrad (taught)
Postgrad (research)

<table>
<thead>
<tr>
<th>Course Type</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Undergraduate</td>
<td>85%</td>
</tr>
<tr>
<td>Postgrad (taught)</td>
<td>8%</td>
</tr>
<tr>
<td>Postgrad (research)</td>
<td>4%</td>
</tr>
</tbody>
</table>

Which of these devices do you own?
What type of phone do you have?

- Mobile phone: 99%
- Tablet: 13%
- eReader (eg. Kindle): 8%
- Laptop: 93%
- Desktop: 28%
- Other: 4%

What brand of phone do you have?

- Apple: 25%
- Blackberry: 12%
- HTC: 7%
- Nokia: 15%
- Samsung: 29%
- LG: 1%
- Sony Ericsson: 8%
- Other: 2%
Campus map: Please rate from 1-5 as the most useful to incorporate into a University Mobile App (1=least useful, 5=most useful)

1: 12%
2: 14%
3: 25%
4: 21%
5: 27%

Course timetables: Please rate from 1-5 as the most useful to incorporate into a University Mobile App (1=least useful, 5=most useful)

1: 3%
2: 2%
3: 5%
4: 14%
5: 76%

Course notes from Blackboard: Please rate from 1-5 as the most useful to incorporate into a University Mobile App (1=least useful, 5=most useful)

1: 3%
2: 5%
3: 13%
4: 17%
5: 62%
**Course discussion forums:** Please rate from 1-5 as the most useful to incorporate into a University Mobile App (1=least useful, 5=most useful)

1  | 19%
2  | 24%
3  | 30%
4  | 16%
5  | 11%

**Exam results:** Please rate from 1-5 as the most useful to incorporate into a University Mobile App (1=least useful, 5=most useful)

1  | 4%
2  | 3%
3  | 10%
4  | 17%
5  | 65%

**Email:** Please rate from 1-5 as the most useful to incorporate into a University Mobile App (1=least useful, 5=most useful)

1  | 3%
2  | 1%
3  | 2%
4  | 10%
5  | 84%

**University contact directory:** Please rate from 1-5 as the most useful to incorporate into a University Mobile App (1=least useful, 5=most useful)
University news & events: Please rate from 1-5 as the most useful to incorporate into a University Mobile App (1=least useful, 5=most useful)

<table>
<thead>
<tr>
<th>Rating</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9%</td>
</tr>
<tr>
<td>2</td>
<td>14%</td>
</tr>
<tr>
<td>3</td>
<td>32%</td>
</tr>
<tr>
<td>4</td>
<td>30%</td>
</tr>
<tr>
<td>5</td>
<td>16%</td>
</tr>
</tbody>
</table>

Verbatim question: frequency of top ten terms.
Frequency was analysed using wordle.net

<table>
<thead>
<tr>
<th>Term</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>timetable</td>
<td>744</td>
</tr>
<tr>
<td>email</td>
<td>708</td>
</tr>
<tr>
<td>blackboard</td>
<td>523</td>
</tr>
<tr>
<td>results</td>
<td>447</td>
</tr>
<tr>
<td>exam</td>
<td>415</td>
</tr>
<tr>
<td>event</td>
<td>287</td>
</tr>
<tr>
<td>map</td>
<td>284</td>
</tr>
<tr>
<td>library</td>
<td>283</td>
</tr>
<tr>
<td>campus</td>
<td>250</td>
</tr>
<tr>
<td>access</td>
<td>227</td>
</tr>
</tbody>
</table>
Appendix II: Buying in, not building: an alternative approach to development

A mini-case study from UCD

Regardless of which approach, app versus web, you still need to decide on who is going to build it or buy it. UCD chose to buy in a platform (Blackboard's Mobile Platform) rather than building one themselves.

The need
They could see the data that there were 30,000 Apple devices (20,000 of which were iOS - iPhones, iPads or iPod Touch). So, there was a clear need to identify and deliver a range of services that would provide a meaningful and valuable smartphone experience.

The approach
Building it in-house was never an option. There are no internal skills or experience of developing mobile apps, so for UCD, it made sense to buy it in.

Another key consideration was the maintenance so a bought-in, supported solution made a lot of sense in that context.

After an evaluation process, Blackboard's Mobile Platform was chosen. It’s a native app for iPhone and Android (though can also be deployed on other platforms like Blackberry or Windows). It offers a single destination for students and staff to access services on their mobiles - see peer analysis section for details. This single destination is also very useful for promotion across campus with simple, clear messages like “Find it on UCD Mobile app”.

Key challenge: data
As we’ve already identified with UCC, one of the key challenges in deploying the UCD app was in data or content access, i.e. having access to the right data in the right format. Many sources of content were able to produce a feed. The site’s content management system, for example, was able to publish RSS feeds for some news and events. This meant that content is produced once and published on the site and within the app, avoiding duplicated effort. Other areas didn’t have a feed. Technically, that was relatively easy to overcome by providing access to a Blackboard interface in which to add content. However, the challenge there was a human one - it can be a struggle to get some areas to update their content. The impact of this is, if content goes out of date, the credibility of the app can really suffer.

“It wasn’t all straightforward or simple, but it was about getting the data right”
Brian Morrissey, UCD
User adoption

The app was released in August 2011 for iPhone (and shortly after for Android) and was downloaded 6,000 times in the first month, which is good total by any standard. 73% of users are on iOS and the most used sections within it are Blackboard Learn, News & Events, closely followed by the campus map and staff directory. As an interesting aside, UCD had previously put a lot of work into their timetable, deploying it to Google Apps which can then be synced down to the native calendar application on the phone. Yet, users still have an expectation, and make requests that to have it within their app.

Roadmap

UCD take a phased approach to delivering on current unmet needs, targeting two app releases a year. In terms of what’s on the roadmap, some of that they’re currently looking are:

1. Places and tours
2. Adding restaurants, computer support to map
3. Parking
4. Grades
5. Library search
6. Web storage
7. Restaurant menu
8. Weather
9. Classified Ads

Benefits

Being locked into a vendor is sometimes seen as a negative, but the benefits for UCD are a predictable cost for the foreseeable future with meaningfully mobile services available to students, delivered at a significant cost savings to a bespoke build.

In addition, with Blackboard’s SDK, UCD still have the opportunity to create their own apps, for example, integration of the college radio station within the app.

Tips from UCD

1. Good governance - their web steering group meets regularly to plan activity for the roadmap and decide what’s in and what’s out.
2. Take a phased approach but prioritise and start with a smaller set of "anchor tenants" or “killer apps”.
3. Reuse existing content whenever possible.
4. Think about things that are quick to consume yet useful and that use the power of the phone - e.g. “Where do I need to be right now?”
5. Get feedback - they use surveys regularly via Google Docs to poll users on their experiences with the app
# Appendix III: Mobile app candidates

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Timetables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What is it</strong></td>
<td>Access to student timetables</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>Students have easy access to their timetables and receive timely updates if there are any changes to lecture. “timetable - updates if timetable are changed. This way student will know straight away if room/time has been changed”</td>
</tr>
<tr>
<td><strong>Considerations</strong></td>
<td>A systems integration project would need to happen first to get the timetable data in a usable format.</td>
</tr>
</tbody>
</table>

*Source: staff survey*
<table>
<thead>
<tr>
<th>Candidate</th>
<th>Timetables</th>
</tr>
</thead>
<tbody>
<tr>
<td>“We have to concentrate on getting fundamentals in order - must get the data in order first before we do anything, not having silos of information in different places. Otherwise we inherit the problems of desktop”. <strong>Source:</strong> stakeholder interviews</td>
<td></td>
</tr>
<tr>
<td>“a mobile app that is actually useful would need to have fully integrated access to the student / staff record systems as well as to geographical and location data. The problems encountered when developing the portal with regards to single sign on and so forth would be the same as those encountered when developing a mobile app. We need underlying systems that can talk to each other before we should spend our time developing an app”. <strong>Source:</strong> Staff survey response</td>
<td></td>
</tr>
<tr>
<td>“If you get the data right you can use a calendar app. Perhaps the project should be this and use phone’s native application”. <strong>Source:</strong> Stakeholder interview</td>
<td></td>
</tr>
<tr>
<td>We don’t believe this is app per se, but rather an integration project involving room bookings and the registrar’s office as key project sponsors.</td>
<td></td>
</tr>
<tr>
<td>And it’s more than mobile. Only once the first piece has been done can the data be useful and used on a mobile, desktop or web applications (like Google Calendar) or native calendar apps.</td>
<td></td>
</tr>
<tr>
<td>Candidate</td>
<td>Campus map</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>What is it</td>
<td>A map with locations and associated data about places (opening times, contact details)</td>
</tr>
</tbody>
</table>
Aids in finding way around, finding buildings, opening times.

“Map of campus with all buildings, lecture halls plus explanations of abbreviations for places”

Source: staff survey.

“Map of the college campus would be particularly useful. Moreover, the inclusion of a searchable directory of staff offices, lecture halls etc would be of great value”.

Source: staff survey

“Boole building, Room 1 is referenced as ‘CK 1’ - building codes are daft”.

Source: stakeholder interview

“Mapping is needed. It’s not just A to B - it’s what services are available using visual communication”.

Source: stakeholder interview.

“What do the building names or codes mean? The Kane versus the Science building, and what’s in their building i.e. labs with the newest machines that are idle, or a restaurant in the basement that no one knows about.”

Source: stakeholder interview.

“University experience doesn’t stop at the campus boundary, it’s also about Cork. You have international students, Dublin, Kerry students, so mapping can extend”
<table>
<thead>
<tr>
<th>Candidate</th>
<th>Campus map</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Considerations</strong></td>
<td>May need a geoDatabase. As the amount of data sources scaled, it would need multiple owners to make sure the data was kept up to date (example: opening times, café menus). What it would need most is one central owner and a clear procedure on how to add new sources of content.</td>
</tr>
<tr>
<td><strong>Our take</strong></td>
<td>What’s nice about this idea is that it need not be delivered all at once, different data sources could be added in a phased approach e.g. Main buildings followed by car park data, followed by off campus data etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Exam Results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What is it</strong></td>
<td>Mobile access to exam results</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>Lets students get fast access to results without need for desktop.</td>
</tr>
<tr>
<td><strong>Considerations</strong></td>
<td>Would probably fit within making the student portal mobile via the web rather than a dedicated app.</td>
</tr>
</tbody>
</table>
## Candidate Exam Results

### Our take

This came out in the student survey as one of the more unanimously rated mobile app opportunities. What should be borne in mind is that the survey went out very close to exam times.

### Course notes (blackboard)

<table>
<thead>
<tr>
<th>What is it</th>
<th>Access to course notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benefits</td>
<td>Students can catch up on notes without having to sit down in front of a desktop.</td>
</tr>
<tr>
<td>Considerations</td>
<td>Blackboard offer an app platform with different cost models (a campus licence versus a user licence where cost is deferred to the student, they pay to download it).</td>
</tr>
</tbody>
</table>

However, there is vendor lock-in in addition to locking out non-iPhone and Android users. Also, what level of integration for the different aspects of the app needs to be explored in more detail.
<table>
<thead>
<tr>
<th>Candidate</th>
<th>University news &amp; events</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is it</td>
<td>News &amp; events listings</td>
</tr>
<tr>
<td>Benefits</td>
<td>Keep students, staff, alumni etc. up to date as to what’s happening at UCC.</td>
</tr>
<tr>
<td></td>
<td>“News feed of up to date things on campus be it a film in the Boole by a society, lunch specials or a lunchtime seminar”.</td>
</tr>
<tr>
<td></td>
<td><strong>Source:</strong> staff survey</td>
</tr>
<tr>
<td></td>
<td>Events and News as this would be a good way of keeping up to date with day to day activities</td>
</tr>
<tr>
<td></td>
<td><strong>Source:</strong> staff survey</td>
</tr>
<tr>
<td>Considerations</td>
<td>While we would need to rely on multiple content sources to provide content for the app, it’s not a huge technical challenge.</td>
</tr>
<tr>
<td></td>
<td>In addition, not all content sources need to be added in one go - they could be added on a phased basis.</td>
</tr>
<tr>
<td>Candidate</td>
<td>Alumni app</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td><strong>What is it</strong></td>
<td>An app which helps keep alumni in touch with each other and events at the university.</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>Alumni are a valuable source of funding for the University.</td>
</tr>
<tr>
<td></td>
<td>Having up-to-date contact information is a challenge - so keeping in touch with them is an important objective.</td>
</tr>
<tr>
<td></td>
<td>“We need the right information to the right people, whether it’s looking at photos of events they attended or they’re moving to Paris, NY, how do they connect to people in the network? We should be linking people together and we really need to get people to update their address &amp; contact details.</td>
</tr>
<tr>
<td><strong>Considerations</strong></td>
<td>A good social media strategy and apps like LinkedIn or Facebook could take care of this.</td>
</tr>
<tr>
<td>Candidate</td>
<td>Access to portal services</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>What is it</td>
<td>Mobile dashboard for access to email, calendar, print top-up, student profile, exam results.</td>
</tr>
<tr>
<td>Benefits</td>
<td>Extending or optimising access to portal from mobile.</td>
</tr>
<tr>
<td></td>
<td>“Student Portal - giving student easiest access possible to their email, blackboard etc - essential services for student”</td>
</tr>
<tr>
<td></td>
<td><strong>Source:</strong> staff survey</td>
</tr>
<tr>
<td>Considerations</td>
<td>Consider responsive design rather than an app as these services are already available online.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Club &amp; Society information</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is it</td>
<td>Updates from clubs and socs</td>
</tr>
<tr>
<td>Benefits</td>
<td>Keep members up to date already</td>
</tr>
<tr>
<td>Considerations</td>
<td>According to the clubs &amp; socs president, this is largely being done already, albeit via Facebook, Twitter and collegeroad.ie</td>
</tr>
<tr>
<td>Candidate</td>
<td>Food menus + restaurants / Bar (Menus / Offers, etc)</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><strong>What is it</strong></td>
<td>What’s for lunch where and what deals are on</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>For the user, to know where to eat.</td>
</tr>
<tr>
<td></td>
<td>“Include info on food on campus in student centre on a particular day or menu in staff rest etc. Opening hours for shops and services”</td>
</tr>
<tr>
<td><strong>Source</strong>: staff survey</td>
<td></td>
</tr>
<tr>
<td><strong>Considerations</strong></td>
<td>Could be a discreet app or a feature/content of a larger app like the campus map/directory - see our peer analysis piece.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Car parking</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What is it</strong></td>
<td>An app which gives live updates of spaces vacant in the different campus car parks.</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>Less congestion, greater utilisation of car parking space</td>
</tr>
<tr>
<td><strong>Considerations</strong></td>
<td>Would need a live feed from the car park service company</td>
</tr>
<tr>
<td>Candidate</td>
<td>Submission Reminders</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------</td>
</tr>
<tr>
<td><strong>What is it</strong></td>
<td>Reminders of when assignments are due</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>Coursework is delivered on time</td>
</tr>
<tr>
<td><strong>Considerations</strong></td>
<td>Fits in student calendar/timetable or within Blackboard</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Library app</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What is it</strong></td>
<td>An app with access to catalogues</td>
</tr>
</tbody>
</table>
| **Benefits** | “Library account access - be able to renew books, etc Check the status of a library book”  
*Source: student survey.*  
“Booleweb section to check for books in library, opening hours and a link to library account for checking when books are due!”  
*Source: student survey.* |
<p>| <strong>Considerations</strong> | There is already a library app available, “Library Anywhere” but there may not be great awareness about it (and it’s not available on every phone). |</p>
<table>
<thead>
<tr>
<th>Candidate</th>
<th>Student voting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What is it</strong></td>
<td>An app which allows students to cast their votes in Students’ Union Elections &amp; Referenda</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>Greater voter turnout</td>
</tr>
<tr>
<td><strong>Considerations</strong></td>
<td>Something to explore with the SU, would need to be very secure.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Grant updates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What is it</strong></td>
<td>An app which updates students where and when they can apply or when to pick up their grant</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>Streamlined grand process. “when local authority grants can be collected from the grants office”</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>student survey</td>
</tr>
<tr>
<td><strong>Considerations</strong></td>
<td>Dependency on a system in the background.</td>
</tr>
</tbody>
</table>
Appendix IV: UCC prototype technical architecture

The prototype components are as follows:

Root folder contains index.html, login.html, robots.txt and the folders described below.

- index.html contains the majority of the pages in the prototype, which jQuery Mobile manages and changes between as necessary.
- A CSS (cascading style sheets) folder ("/css") containing a single CSS file which covers everything in the demo.
- An Images folder ("/img") contains any image resources for the demo.
- A script folder ("/js") containing jQuery and jQuery mobile libraries, as well as the main JavaScript resource ("main.js"). Both jQuery and jQuery Mobile are downloaded from the jQuery CDN at runtime.
- Secure folder ("/secure") containing the secure logged in pages of the app, such as "My Schedule".
- An XML folder ("/xml") which contains any content for the prototype, such as facilities, staff directory, student services. These are parsed by javascript on the fly and are the only source of content. There is no database or content from anywhere else apart from Google Maps.

The prototype uses jQuery mobile to manage the navigation between pages and provide a consistent UI across devices. jQuery Mobile also uses a CDN hosted css file which sets up some default jQuery Mobile styles.

While the user is navigating between screens the browser stays on the same html page but shows different sections of it for each mobile screen. This allows for the use of
JavaScript to persistently store data. Any stored data will be dropped if a page is refreshed or if the user navigates to another html page. This doesn't cause any issues for the purpose of the demo, and allows listings and details pages to dynamically show the correct content based on the users chosen path.