UVic Access

Campus Mobile Application Design Feasibility Report

Team Pandas:

David Clark, Adnan Duale, Kushal Patel

April 2, 2016

Team Pandas

123 Fake Street

Victoria, BC V6R A7J

April 2, 2016

Ann Onymous, App Development Coordinator

University of Victoria, Computer Science Department

3800 Finnerty Road

Victoria, BC V8P 5C2

Salutations:

In response to our initial agreement regarding RFP:1 (A Smartphone App for Students of the University of Victoria), please accept the attached report. Team Pandas herein details a comprehensive proposal for the design of a mobile application to replace the current UVicMobile software developed by Apparmor, Inc.

This report finds that the current app, UVicMobile, has failed to be adopted by students due to its unfriendly design and lack of features. We use user feedback and a review of current best practices, as well as comparison with class-leading apps in other parts of North America, to inform an app design that provides students with features that will ensure adoption. We also implement in prototype form a user interface that overcomes the main objections to the use of the previous app. This report also advocates and describes proven techniques in software development and testing, and thereby promotes software reliability and contributes to low life-cycle costs. We thereby demonstrate the conceptual feasibility of a world-class replacement app, one in line with UVic’s public messaging on technological pre-eminence.

Thank you for your time and your generous consideration of our proposal. We look forward to working with you on this exciting project.

Sincerely,

David Clark

Adnan Duale

Kushal Patel

Team Pandas

Attachment: UVic Access

Table of Contents

[List of Figures And Tables iv](#_Toc447456768)

[Executive Summary v](#_Toc447456769)

[Introduction 1](#_Toc447456770)

[Discussion 1](#_Toc447456771)

[1. Alternatives Considered 1](#_Toc447456772)

[2. Features of the Current App 2](#_Toc447456773)

[2.1 Methodology 2](#_Toc447456774)

[2.2 Omitted Features 4](#_Toc447456775)

[3. Top Six Universities Mobile App Research 5](#_Toc447456776)

[4. Recommended Design 6](#_Toc447456777)

[4.1 Design Layout 7](#_Toc447456778)

[4.2 Brief Overview 13](#_Toc447456779)

[5. Development Plan 14](#_Toc447456780)

[6. Assuring Software Quality 15](#_Toc447456781)

[Conclusion 17](#_Toc447456782)

[Recommendations 18](#_Toc447456783)

[References 19](#_Toc447456784)

[Appendicies 20](#_Toc447456785)

[Appendix A: Work Logs 20](#_Toc447456786)

# List of Figures And Tables

[Table 1. UVicMobile Features 2](#_Toc447376521)

[Figure 1. Fully implemented weather page 3](#_Toc447376522)

[Figure 2. Minimally implemented map page 3](#_Toc447376523)

[Table 2. UVicMobile features – degree of implementation 4](#_Toc447376524)

[Figure 3. CSUN Mobile App [8]. 5](#_Toc447376525)

[Figure 4. UVIC Access startup 6](#_Toc447376526)

[Figure 5. UVIC Web Services 7](#_Toc447376527)

[Figure 6. Netlink Login 8](#_Toc447376528)

[Figure 7. Connex Login 8](#_Toc447376529)

[Figure 8. Uvic My Page 8](#_Toc447376530)

[Figure 9. Registering for a course 9](#_Toc447376531)

[Figure 10. Confirm Registration 9](#_Toc447376532)

[Figure 11. View Grades 10](#_Toc447376533)

[Figure 12. UVic Coursespaces 10](#_Toc447376534)

[Figure 13. UVic Webmail 11](#_Toc447376535)

[Figure 14. UVic Connex 12](#_Toc447376536)

[Figure 15. Connex submissions 12](#_Toc447376537)

[Figure 16. Maps. 13](#_Toc447376538)

[Figure 17. Building Maps. 14](#_Toc447376539)

[Figure 18. Agile Development Software Lifecycle [9]. 15](#_Toc447376540)

[Table 3. Software Qualities [10] 16](#_Toc447376541)

# Executive Summary

This report contains a new design for a replacement for the current UVic Mobile app. The current app is poorly reviewed and adopted by UVic students due to its lack of features, inconsistent design and implementation, and poor user interface. The new design included in this report addresses these weaknesses by identifying desirable features; ones that are both in demand by UVic students, and have been incorporated by similar apps available at other post-secondary institutions.

In addition to feature and user-interface proposals, this report suggests a series of development and testing methodologies that will ensure the long-term viability of the application, support its quality, and encourage its positive reception among students.

This design is especially timely since the Apparmor-UVic contract expires this month; adopting this design will present an opportunity to provide UVic students with a world-class mobile app, driven by the latest principles of app design and development. Such an app will serve as a powerful inducement to enrolment, and will raise the statue of UVic as a home of technologically sophisticated software development, in line with the new Edge public messaging.

# Introduction

In 2014, the University of Victoria enshrined the goals of technological excellence, student empowerment and dynamic support of campus life in its guiding principles, the Edge branding guidelines [1]. At the same time, UVic released UVicMobile, an application designed to be used by students on tablets and other mobile devices. The designers of this application (or ‘app’) intended it to be a popular, effective information support technology for student use. Unfortunately, UVicMobile proved to be a failure – three years after its release, few students have downloaded it, and poor reviews of the app reflect frustration with its lack of features and sloppy implementation [2].

Addressing these concerns requires a redesign of the app. This study examines the feasibility of different approaches to implementing missing or under-realized features, and correcting the poor user-interface design of the original app. The process of demonstrating the feasibility of a new design will is a four step process:

1. Evaluate the features of current app, identifying strengths and weaknesses.
2. Examine ‘best-in-breed’ apps available to other institutions.
3. Assemble a list of needed features from the above analysis.
4. Demonstrate that a better, modern user interface is possible by creating an interface mock-up.

# Discussion

## 1. Alternatives Considered

A third party design, implemented by the University as an official campus app, is the most feasible conceptual approach to correcting the current situation.

The alternative of modifying the current app is not available, as the current app was developed by Apparmor [3], a Toronto-based software development company, and the rights to this app revert to them upon the conclusion of their 3-year contract with UVic.

A second approach is available – a 3rd-party, unofficial app could be created, that would source its information directly from the publicly-available UVic website [4]. This approach, while possible, would be technically and practically infeasible; even if UVic tolerated the existence of an unofficial app competing with theirs, a 3rd-party app that lacked access to UVic’s information technology through a proper Application Programming Interface (API) would have to resort to ‘screen scraping’, an unreliable and maintenance-intensive form of data access.

An official app that incorporates design features as outlined below is therefore the best alternative – it would enjoy the straightforward access to data required for a low-cost, low-maintenance implementation, and could use UVic branding, marketing and support resources without restriction.

## 2. Features of the Current App

The feature analysis begins by considering the feature set of the current app as a baseline set; if the poor adoption of the UVicMobile app is driven in part by the perception that it lacks features, the current features should be considered to be the minimum acceptable.

The features of the current app are depicted in Table 1, functionally divided into groups based on their primary role. Static information functions display data that rarely changes – campus maps, service addresses and contact information, and links to general internet resources such as social media. Dynamic information functions are updated weekly, daily or on-demand, and leverage the advantages of an electronic medium over more traditional presentation methods such as brochures or newsletters. Safety tools are not informational as such; they rather provide certain actual capabilities to the user such as the activation of a mobile phone’s flashlight feature (iPhone only), or direct-dialling campus security.

Table . UVicMobile Features

|  |  |  |
| --- | --- | --- |
| **Static information** | **Dynamic information** | **Safety** |
| * Campus map * Campus dining * Social media * Security * Emergency info | * Events Calendar * Transit * Weather | * Safety tools |

### 2.1 Methodology

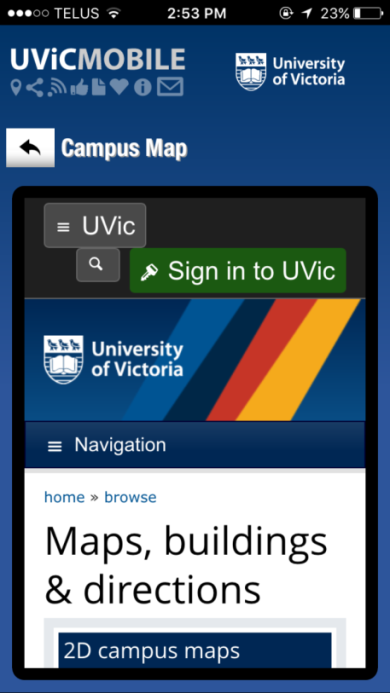
The mere appearance of a feature in Table 1 does not imply that the feature itself is satisfactorily implemented in the app – many features are present in a form that makes them more difficult to use within the app than if they were accessed by the mobile device’s other applications. To establish which features are fully usable, and which are present in a merely perfunctory sense, the features and sub-features of UVicMobile are divided in Table 2 into two categories, ‘Fully Implemented’ and ‘Minimally Implemented’.

Fully Implemented (FI) features are those that have been designed with the strengths and limitations of a mobile device in mind. An example is depicted in Figure 1 – navigation is performed using large buttons, easily selected on a small touch screen. The textual information on the page is formatted appropriately for a small screen; text is centred and large enough to read easily. No zooming and minimal scrolling are necessary to use the feature.

Figure . Fully implemented weather page



Figure . Minimally implemented map page



Minimally implemented features are those that, by contrast, have been included in the app without regard for the particular limitations of mobile platforms. Figure 2 depicts the current UVicMobile app’s map display – it is a simple reproduction of the corresponding UVic website’s map page [5]. This page was designed to be viewed on a full-sized desktop monitor, and interaction is based on keyboard and mouse input. Navigation of this page depends on finding small links which take the form of ordinary text, rather than clear, obvious action buttons. Extensive scrolling is necessary to view the information the page provides, and zooming is mandatory due to the small screen size relative to the total amount of information on the page.

Table 2 summarizes the distribution of sub-features between these two categories. Nearly half the features included in the current app fall into the ‘minimally implemented’ category. This situation reflects the development philosophy of Apparmor, whose business strategy depends on the production of large numbers of very similar apps for a variety of post-secondary institutional customers, at a minimum possible cost [6]. A modern, usable and successful app must fully implement the vast majority of its features; decisions to eschew full implementation must be driven by usability, not cost or convenience.

Table . UVicMobile features – degree of implementation

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Feature** | **Full** | **Minimal** |
| Events | Today’s Events, This Week’s Events, This Month’s Events | X |  |
| Events | Date, Category, Description, Time, Pricing and Location | X |  |
| Events | Search by category | X |  |
| Campus Map | 2d, 3d accessibility and parking |  | X |
| Transit | A list of next 4-5 buses leaving UVic for each route, with times and bay number | X |  |
| Weather | Environment Canada feed for upcoming week | X |  |
| Campus Dining | Opens the corresponding section of the UVic website |  | X |
| Social Media | Opens browser views of UVic’s Facebook, Youtube, Youtube (Vikes), Flickr and 4 different twitter accounts |  | X |
| Security | Buttons to call 911 and campus security | X |  |
| Security | Browser views of campus security location map and website |  | X |
| Safety Tools | Loud alarm, flashlight (iPhone only), email your location, security alerts | X |  |
| Other | Link to UVic website |  | X |

### 2.2 Omitted Features

The features in Table 2 clearly don’t include all possible and desirable features of a campus mobile app. Analysis of the reviews of the current UVicMobile app [2] reveals recurrent requests for access to a variety of features:

* UVic Mypage
* Connex
* Coursespaces
* Webmail
* Grades and Registration
* Course Information

This finding was echoed by research team brainstorming and informal discussion with potential users of a replacement app.

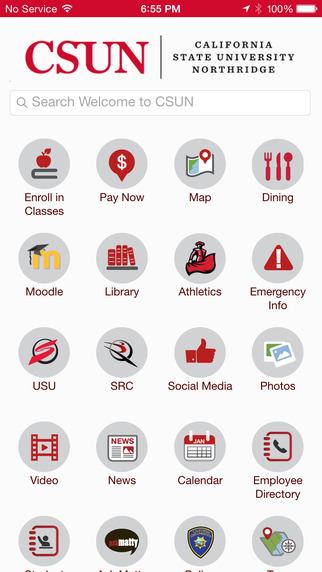
## 3. Top Six Universities Mobile App Research

After research we found that many universities, both large and small, have great mobile campus experience, but the top six universities or colleges who created most appealing mobile apps are the following:

* Barnard College: Virtual Tour
* California State University Northridge (CSUN): Indoor Maps
* Qatar University (QU): Multiple Languages
* Fitchburg State University (FSU): IT Help
* University of Massachusetts (UMASS): Notification and Banner Messages
* Georgetown University: Event of the week.

Offering engaging mobile applications that provide tours of the university, such as Barnard College leaves the application user with a real life experience [7].  For students who are considering attending a university but do not have the funds to do a real life visit, mobile apps like this provide the best type of marketing. The university has the ability through the app to provide the possible a virtual tour of their campus. As shown in Figure 3, the California State University takes it a step further by providing students with maps that showcase where information such as the bathrooms, classrooms, water fountains are.  All these features help students relate with the university and help in creating familiarity with the campus, which is what the university is trying to accomplish.

Figure . CSUN Mobile App [8].



Universities are also using apps to collect feedback and engage with students, staff and alumni.  For instance, Qatar University app includes a section about the campus project and where staff can leave their feedback and ideas. The app is also available in two languages, allowing the users to pick which language they are the most comfortable in using [7]. This is great for students as they can provide their input right from their phone making it easy and convenient and in the language of their choice.

Fitchburg State University has made it easier for their students to get IT help [7]. This school apps allows students to click and chat to receive IT help right from their phone. University more and more are using apps to communicate with their students.  University of Massachusetts also uses push notification to remind students of what they need to bring during certain events such as orientation, career fairs and exams, and provide detailed instructions [7].  UMass uses their app to communicate with their students on the process of move in. The app includes important information, such as dates, times, as well as detailed information and maps, making it easier for the students and families during the hectic period.  Georgetown University has included a feature for students which highlights what is going on the campus for that week such as basketball games and debates [7].

## 4. Recommended Design

Figure . UVIC Access startup

main icon.png

Based on our research and system analysis of the current UVIC Mobile application, we recommend an appropriate redesign of the app to incorporate some key features that the current app lacks. It is a touch screen application suitable for iOS as well as Android. This is not a complete design and requires further development which will be addressed in the next section.

Figure 4 shows the home screen icon and the basic start up to the first screen of the design.

### 4.1 Design Layout

This key feature of the design lets students access the different online services offered by UVIC through their cell phones.

Figure . UVIC Web Services



As seen in Fig 5, UVIC provides the following web services:

* MyPage
* Coursespaces
* Webmail
* UVIC Connex

Once the user taps UVIC Web Services from the home screen, they are asked to login with their UVIC netlink id and password as seen in Figure 6. This gives them access to the first three services. To access connex, students have to login once again using their connex login and password as shown in Figure 7.

Figure . Netlink Login



Figure . Connex Login



MyPage lets student access all the features available on their desktop site (My Home, Student Services and Research). Fig 8 shows the usage of tabs to give quick access to these features.

Figure . Uvic My Page

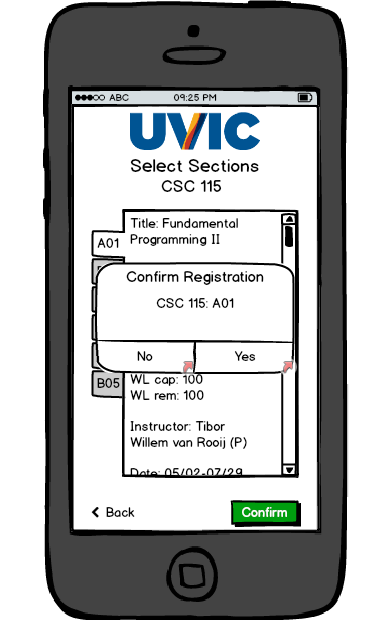
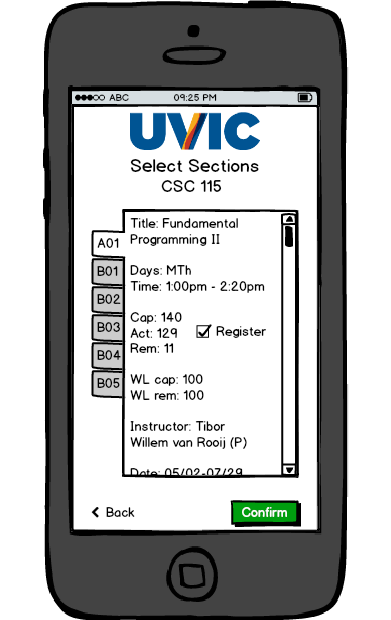
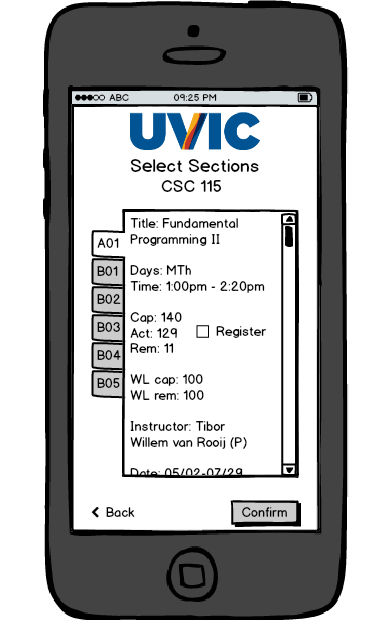


Figure . Registering for a course



MyPage lets student register for courses. Figure 9 illustrates all the options available while registering for courses. Drop down menu is used to select desired term and all available courses are listed below. Selecting the desired course and section prompts the user for confirmation before successfully registering them in the course (illustrated in Figure 10).

Figure . Confirm Registration



As seen in Figure 10, Section of each course are listed in tab format. Selecting the desired tab, and checking the register box allows the user to ‘Confirm’ their registration.

Additionally, this tab format is used to display grades as well. A drop down menu is used to select the term and tabs display each course for the particular term as seen in Figure 11.

Figure . View Grades

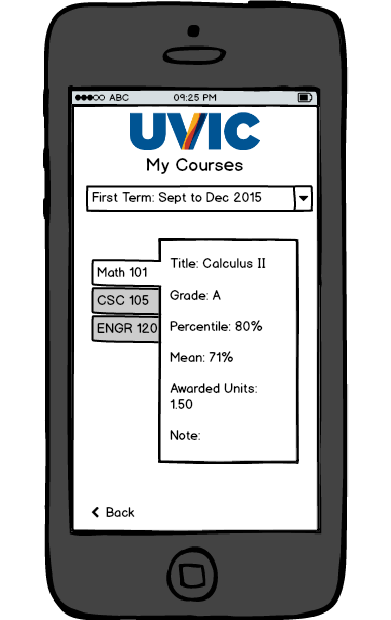
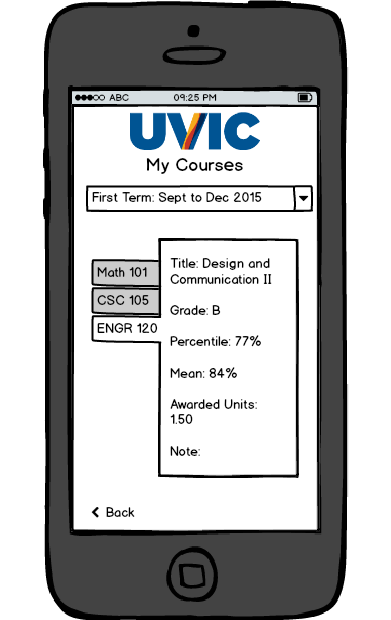
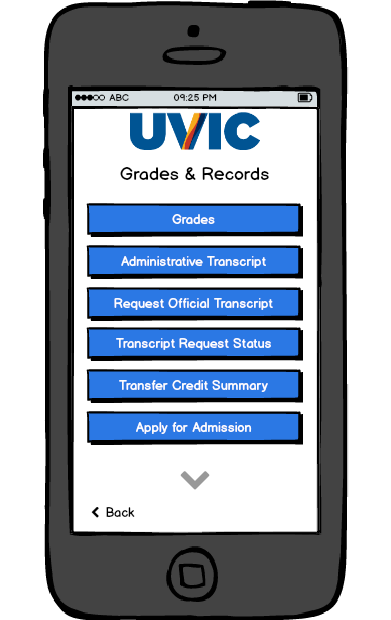


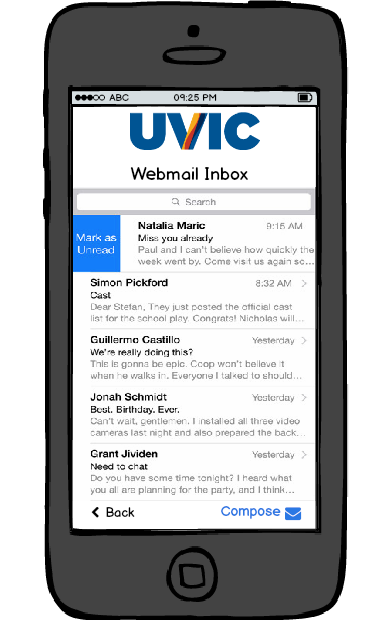
Figure . UVic Coursespaces

****

UVIC Coursespaces uses similar style with tabs to ease navigation as well as display all required information. Figure 12 displays the incorporation of Coursespaces in our design.

UVIC provides an email service called Webmail which is also a part of our design. Figure 13 displays layouts for the menu, inbox and composing new mails for this feature.

Figure . UVic Webmail



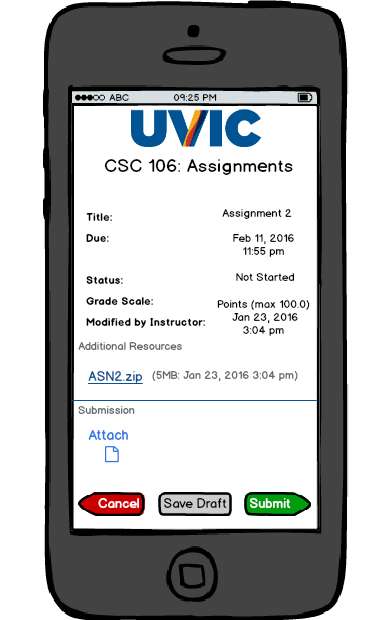
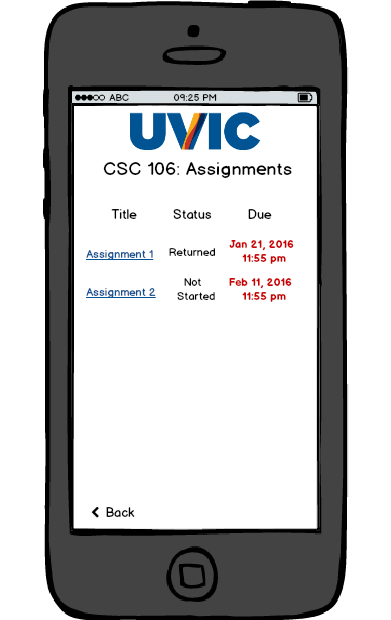
UVIC Connex is a service used mostly by their Engineering students. It is very similar to UVIC Coursespaces, but requires its own separate login via CAS (Figure 7). Figure 14 lays out a design to adopt Connex for cell phones.

Figure . UVic Connex



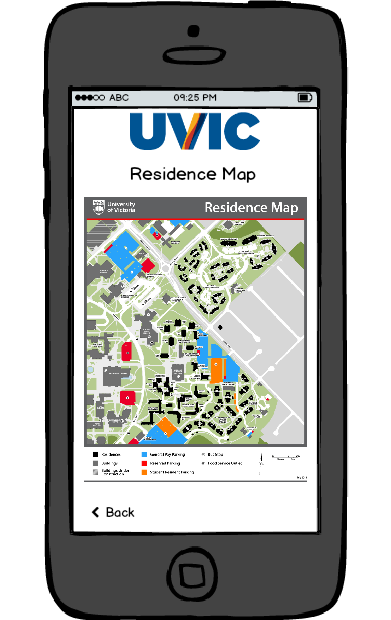
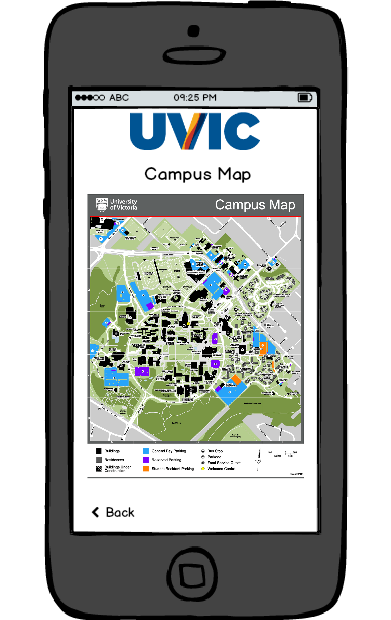
Connex also lets users submit assignments. This is done by simply attaching the desired attachment on the submission screen and clicking ‘submit’ as shown in Figure 15.

Figure . Connex submissions



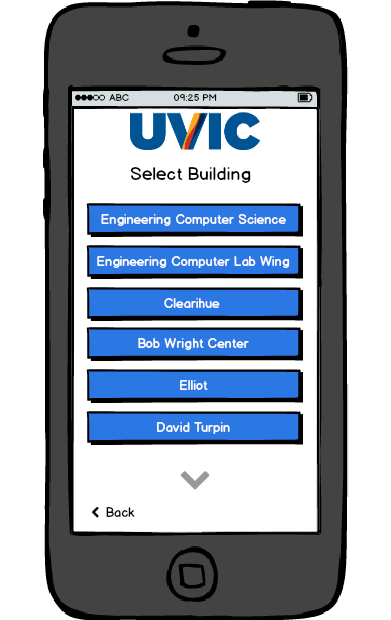
The ‘Maps’ feature is also redesigned and users can view maps within the app itself, rather than redirecting the user to the UVIC website. Figure 16 shows this new layout.

Figure . Maps.



In addition to the campus map, maps of residence and buildings are added as well. Each building is mapped according to the floor as seen in Figure 17.

Figure . Building Maps.



### 4.2 Brief Overview

Some features were added to our design while some were adopted from the current UVIC Mobile app. Here is a list of all the features required in our design.

New Features:

* My Page - allow full access to UVIC MyPage Service
* Coursespaces - allow full access to UVIC Coursespaces
* Webmail - allow full access to UVIC Webmail
* Connex - allow full access to UVIC Connex
* Maps - Integrate UVIC maps in the software

Adopted Features

* Upcoming Events - lists all upcoming events at UVIC
* BC Transit - lists schedules of all the buses departing UVIC within the next half hour
* Safety and Emergency - lists all emergency contacts at UVIC, allows toggling of flashlight and alarm and provides the ability to directly call 911.
* Campus Dining - lists the location and timing of each cafe on UVIC along with their menus

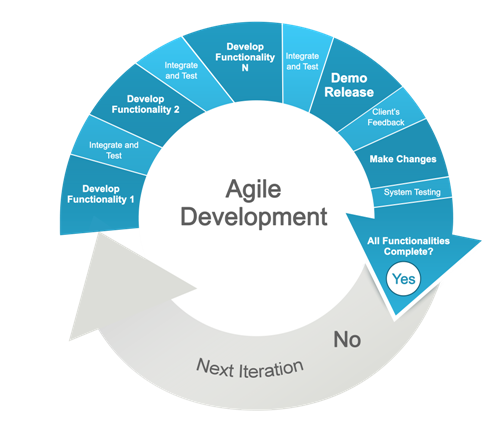
## 5. Development Plan

This is a pretty basic implementation of a mobile application. Layouts are already provided, and further testing, evaluation and possibly redesign is required. This application implements features that are already designed by UVIC in web format to a cell phone format.

This project requires a team of at least 4 members:

* Leader: group leader, coordinate meetings, look over project flow and guide the rest of the team
* Analyst: analysis current design and further iterations, performs various tests and provides feedback to the designer.
* Designer: designs the look of the app based on the layout provided over multiple iterations
* Programmer: handles the coding and networking

Figure . Agile Development Software Lifecycle [9].



This project requires multiple iterations of testing and evaluation and therefore the **Agile** model is the recommended plan to follow. These reiterations also allow developers to add additional features at any stage before release.

This model implies that each feature must be implemented, tested, evaluated, reiterated if necessary and then integrated before any other feature, which follows the same suit.

## 6. Assuring Software Quality

This is a project that needs to function very well upon launch and releasing beta versions (semi-implemented) can hurt the app. Therefore, testing and evaluating is very important.

Some recommended test are listed below:

* Software Verification: Verify whether developed app meets initial requirements
* User Experimentation: Test navigation of the app by interviewing people and asking them to perform use cases on the app.
* Software Validation: Validate that app meets all requirements
* Software Quality Assurance: Fill Table 4.1 to assure quality.

Additional tests may be required as the analyst of the team performs testing and gets feedback on which he can reiterate and test again.

Table . Software Qualities [10]

|  |  |
| --- | --- |
| **Software Quality** | **UVIC Access** |
| Understandability  Are variable names descriptive of the physical or functional property represented? Do uniquely recognisable functions contain adequate comments so that their purpose is clear? Are deviations from forward logical flow adequately commented? Are all elements of an array functionally related? |  |
| Completeness  Are all necessary components available? Does any process fail for lack of resources or programming? Are all potential pathways through the code accounted for, including proper error handling? |  |
| Conciseness  Is all code reachable? Is any code redundant? How many statements within loops could be placed outside the loop, thus reducing computation time? Are branch decisions too complex? |  |
| Portability  Does the program depend upon system or library routines unique to a particular installation? Have machine-dependent statements been flagged and commented? Has dependency on internal bit representation of alphanumeric or special characters been avoided? How much effort would be required to transfer the program from one hardware/software system or environment to another? |  |
| Consistency  Is one variable name used to represent different logical or physical entities in the program? Does the program contain only one representation for any given physical or mathematical constant? Are functionally similar arithmetic expressions similarly constructed? Is a consistent scheme used for indentation, nomenclature, the color palette, fonts and other visual elements? |  |
| Maintainability  Has some memory capacity been reserved for future expansion? Is the design cohesive—i.e., does each module have distinct, recognizable functionality? Does the software allow for a change in data structures (object-oriented designs are more likely to allow for this)? If the code is procedure-based (rather than object-oriented), is a change likely to require restructuring the main program, or just a module? |  |
| Testability  Are complex structures employed in the code? Does the detailed design contain clear pseudo-code? Is the pseudo-code at a higher level of abstraction than the code? If tasking is used in concurrent designs, are schemes available for providing adequate test cases? |  |
| Usability  Is a GUI used? Is there adequate on-line help? Is a user manual provided? Are meaningful error messages provided? |  |
| Reliability  Are loop indexes range-tested? Is input data checked for range errors? Is divide-by-zero avoided? Is exception handling provided? It is the probability that the software performs its intended functions correctly in a specified period of time under stated operating conditions, but there could also be a problem with the requirement document |  |
| Efficiency  Have functions been optimized for speed? Have repeatedly used blocks of code been formed into subroutines? Has the program been checked for memory leaks or overflow errors? |  |
| Security  Does the software protect itself and its data against unauthorized access and use? Does it allow its operator to enforce security policies? Are security mechanisms appropriate, adequate and correctly implemented? Can the software withstand attacks that can be anticipated in its intended environment? |  |

**Note:** Fill out Table 3 only after integrating all features and during the final testing phase

# Conclusion

The unpopularity of the current UVicMobile app and the poor reviews it has received may create the impression that implementing a compelling set of features in a campus mobile app is impossible. The poorly-implemented user interface of the current app may likewise create the suspicion that it is impossible to create a modern-looking, usable and responsive app. Our spec design shows that these fears are unfounded – there are no conceptual barriers to the creation of a replacement app that combines compelling features with a world-class user experience.

This new design, once implemented, will provide UVic students with the access to campus information resources such as Connex, Coursespaces and Webmail, and will do so in a fully-implemented, native manner consistent with strict principles of mobile app design. This design will also yield an app that is equivalent in quality to apps recognized as among the very best; a status currently only available to apps from schools such as California State University Northridge and Qatar University. Finally, this design will leverage the most effective software development and testing methodologies to ensure that UVic students receive the digital support commensurate with the new UVic Edge philosophy.

# Recommendations

To implement the replacement app described in this report, the University should enact the following:

1. Carry through with the cancellation of the current Apparmor maintenance contract.
2. Discontinue efforts to develop a replacement app with outside development companies (if any).
3. Adopt the feature list and UI design herein as the official, supported UVic Mobile application.
4. Retain developers and other associated resources as deemed appropriate by the University.

# References

[1] University of Victoria. “Why the Edge?” [www.uvic.ca](http://www.uvic.ca). [Online]. Available: <https://www.uvic.ca/brand/about/index.php> . [Accessed: March 30, 2016].

[2] Google. “UVic Mobile” play.google.com. [Online]. Available: <https://play.google.com/store/apps/details?id=com.cutcom.apparmor.uvic&hl=en> . [Accessed: March 30, 2016].

[3] Apparmor. “Apparmor” apparmor.com. [Online]. Available: <http://apparmor.com/> . [Accessed: March 30, 2016].

[4] University of Victoria. “Home” [www.uvic.ca](http://www.uvic.ca). [Online]. Available: <http://www.uvic.ca/> . [Accessed: March 30, 2016].

[5] University of Victoria. “Maps, buildings and directions.” [www.uvic.ca](http://www.uvic.ca). [Online]. Availabile: <https://www.uvic.ca/home/about/campus-info/maps/index.php> . [Accessed: March 30, 2016].

[6] Apparmor. “Apparmor – Pricing” apparmor.com. [Online]. Available: http://apparmor.com/pricing/large/ . [Accessed: March 30, 2016].

[7] Modo Labs Team (2015, December 17). [Online]. Available: <http://www.modolabs.com/blog-post/the-10-best-university-mobile-apps-of-2014/>. [Accessed: March 18, 2016]

[8] iTunes AppStore. (2015, November 6). CSUN. [Online]. Available: https://itunes.apple.com/us/app/csun/id677332861?mt=8. [Accessed: March 18, 2016]

[9] Sam Joseph. (2012, October 8). “The Reformation of Software Engineering Process: Agile Lifecycle” shohe-i.blogspot.ca. [Online]. Available: <http://shohe-i.blogspot.ca/2012/10/the-reformation-of-software-engineering.html> . [Accessed: April 2, 2016]

[10] Wikibooks. (2015, May 1). “Introduction to Software Engineering/Quality” wikibooks.org. [Online]. Available: <https://en.wikibooks.org/w/index.php?title=Introduction_to_Software_Engineering/Quality&oldid=2951733> . [Accessed: April 2, 2016]

Balsamiq Mockups 3 (<https://balsamiq.com/products/mockups/>) was used to provide the prototype and screens for this report.

# Appendicies

## Appendix A: Work Logs

David Clark

|  |  |  |
| --- | --- | --- |
| **Date** | **Task** | **Time** |
| Feb 24 | Team formation, first meeting | 30 minutes |
| March 2 | Task analysis | 15 minutes |
| March 5 | Initial point-form draft of current app | 30 minutes |
| March 5 | Survey of online reviews of current app | 15 minutes |
| March 6 | Research into background of current app | 30 minutes |
| March 7 | Interview with Student Services | 1 hour |
| March 7 | Report Template | 20 minutes |
| March 12 | Planning meeting at UVic – outside class time | 90 minutes |
| March 16 | Team presentation outline | 45 minutes |
| March 20 | UVic Mobile app background section – 1st draft | 2 hours |
| March 27 | Presentation meeting at UVic – outside class time | 2 hours |
| April 1 | Section 1 & 2 writing | 2 hours |
| April 2 | Collated contributions from other members. Wrote front and back matter | 5 hours |
| April 3 | Edits for consistency. Work logs. | 1 hour |

Kushal Patel

|  |  |  |
| --- | --- | --- |
| **Date** | **Task** | **Time** |
| Feb 24 | Milestone 1 | 30 minutes |
| Mar 2 | Research: Software Qualities | 1 hour |
| March 3-5 | Research: Software cycles | 3 hours |
| March 9 | Milestone 3 | 30 minutes |
| March 10-13 | Designing prototype | 6 hours |
| March 14 | Milestone 4 | 30 minutes |
| March 16 | Milestone 5 | 30 minutes |
| March 23 | Team Presentation preparation | 1 hour |
| March 23-30 | Designing prototype | 12 hours |
| March 30-Apr 2 | Final Report | 5 hours |

Adnan Duale

|  |  |  |
| --- | --- | --- |
| **Date** | **Task** | **Time** |
| February 24 | Milestone 1: Met with teammates to build Team guidelines document and submitted. | 30 minutes |
| March 02 | Met with teammates to do the Task Analysis, Work plan, Collaboration Strategy and Team Guidelines | 30 minutes |
| March 5 | Evaluate/study current UVic mobile app and research the characteristics of the best University mobile Apps. | 1.5 hours |
| March 07 | Milestone 3: Met with teammates to build the Report Template. | 30 minutes |
| March 12 | Milestone 4: Met with teammates( comp lab) to build the Progress Report. | 30 minutes |
| March 16 | Milestone 5: Met with teammates to build the Team Presentation Template. | 30 minutes |
| March 17-18 | More research on top university mobile apps, described most appealing features that UVic app lacks and what it needs, worked on the introduction, team tasks and methodology part on the project. | 5 hours |
| March 23 | Met with teammates to build the Team Presentation Preparation. | 1 hour |
| March 30-Apr 3 | Worked on the Final project report. | 5 hours |