



Strut Final Capstone Document

Capstone Project Assignment 5

Abstract

Strut? A mobile app in development. The following document gives you a brief look into 5 weeks of blood, sweat and tears that went into this development project.

Group 6

Members

- Jamie Lee Van Der Berg
- Khuliso Sikhwivhulu
- Dahraan Abrahams
- Group 6aggie
- Areeb Royeppen
- Dean Ockhuizen
- Moegamat Ismaeel Ed'rees Jefferies

Sections of Changes

Human Computer Interaction

- [Conceptual Models](#) (Changes)
- [Storyboard](#) (Changes)
- [Maker Movement](#) (Changes)
- [Standard Task](#) (Changes)
- [Consent Forms](#) (Changes)
- [User Interviews](#) (Changes)
- [Field Study](#) (Changes)
- [Controlled Study Vs Field Study](#) (Changes)
- [Overview Process](#) (Added)
- [Comparison Table](#) (Changes)

Software Engineering

- [Process Model](#) (Added)
- [Engineering Principles that guide Process](#) (Added)
- [Software Engineering Principles that guide Practice](#) (Added)

Database

- [Internal Model in MySQL](#) (Changes)

Contents

Human Computer Interaction	5
Capstone Assignment 1	5
Documentation on how we collected the users' needs/requirements for this project	5
User Profiles and Scenarios	7
Hierarchical Task Analysis	9
Requirements	12
Capstone Project Assignment 2	17
Conceptual Models	17
Storyboard Prototype	20
Cardboard Prototype	21
Sketch of applications main screen	22
Is choice Usability or User experience	24
Experience map	25
New interaction issues	25
How does your app relate to the Maker Movement?	26
What is the role of SDKs in this regard?	26
Assuming AgileUX, what goes into Cycle 0?	27
Prioritise the requirements	28
Sketch 4 cycles e.g. Cycle 0 and Cycles 1-3	29
Mock-up of landing page (Horizontal)	30
Prototype with Design Patterns	30
Capstone Assignment 3	32
Standard Task	32
Consent Form	34
User Interviews	35

Field Study	41
Controlled Study vs Field Study	43
Capstone Assignment 4	46
Heuristic Evaluation	46
Consent Form	46
Overview Process.....	47
Evaluation Period	48
Redesign, what has changed and Why	52
Comparison to Previous Test.....	54
Comparison Table	56
Software Engineering.....	57
Process Model	57
General Principles	57
Goals	58
Umbrella Activities.....	58
Engineering Principles that guide Process.....	60
Software Engineering Principles that guide Practice.....	60
UML Diagram	60
Databases.....	61
Entities	61
Business Rules.....	61
ERD Diagram	62
Internal model in MySQL, data types for attributes.....	63
Appendix	66
Appendix A.....	66
Appendix B.....	68
Appendix C.....	69

Appendix D..... 70

Appendix E 71

Appendix F 72

Human Computer Interaction

Capstone Assignment 1

Documentation on how we collected the users' needs/requirements for this project

Interview (one-on-one)

This is one of the most common gathering techniques which allows us to understand what the stakeholders like about the current method of finding venues as opposed to the one which is to be created. It allows us to understand the concerns that the stakeholders might have regarding the new mobile application. [Interview](#) gathering techniques allow us to know which improvements to make to the method of find venues when creating the mobile application and what features stakeholders would find very useful in the mobile system that are not in the current system.

Links to:

Interview questions –

https://drive.google.com/file/d/1tu0bCIQ1N_YOzrIIDMkQLyaX-ga6SrnO/view?usp=sharing

Interview recordings –

<https://drive.google.com/drive/folders/1gYqiffwv5Xeq2mTTOWstYzf5JlyJCUCv?usp=sharing>

Conducting a survey

Conducting survey can produce qualitative and [quantitative data](#). An online survey provides the opportunity to reach out to a larger and diverse group. This also allows us to clear up the software requirements for the developers. As of the tying of this there is over 30 responses that filled out the survey.

Live online survey (Depending on what you respond for a specific question it will take you to a specific section)-

https://docs.google.com/forms/d/e/1FAIpQLSeGME2NEEZRfj6Bpvz1stVaSsBUlvXz2iWm3ISJx_VATOJE7w/viewform?usp=sf_link

Researching similar mobile applications

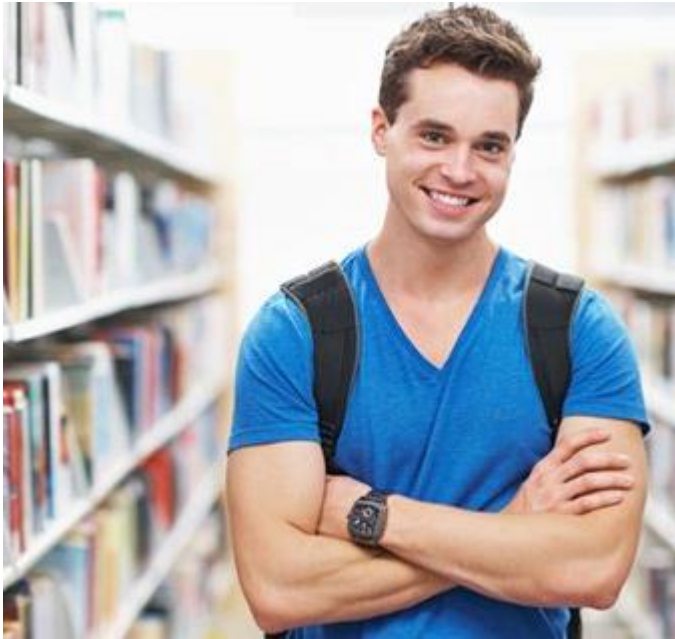
This is used as a starting point of the project where we look at similar products that is already being used. This gives us a basis and an example of how the mobile application should work and information about the work that that needs to be done for our mobile application for it to meet its objective. Reviews allow for use to think of what feature the user would like and allows us to try and incorporate those features to improve functionality for the users

Researching similar systems

Apps	Description
Google Maps	Web mapping service developed by Google in 2005
Apple Maps	Web mapping service developed by Apple Inc. in 2012
City Mapper	Public transit and mapping service developed by City Mapper Limited in 2011
MapQuest	Origins date to 1967 with the founding of Cartographic Services, a division of R.R. Donnelley & Sons in Chicago, Illinois, which moved to Lancaster, Pennsylvania, in 1969

User Profiles and Scenarios

Profile 1



Name: Willem Steenkamp

Age: 20 years' old

Gender: Male

Race: White

First Language: English

Studying LLB at UWC (First Year)

Only child

Uses Android device

About Willem

- Forms part of many campus societies
- Has a long list of modules
- Drives to and from campus

Motivations

- Helping the less fortunate
- Keeping in shape
- Sense of accomplishment

Frustrations

- Being swamped with work
- Inability to remember song lyrics
- Being lost

Scenario 1

As a first-year student, Willem struggles to navigate through the large UWC campus. He often forgets which paths to take to get him from one class to another. With the large number of lectures, he has in a week, including the society workshops he attends, he finds it difficult to make it to classes or extra

murals on time. For Willem, being lost is a massive frustration and inconvenience. As such, he would be less anxious if there were a way for him to easily find venues or buildings without the burden of remembering everything.

Profile 2



Name: Thembile Mafu

Age: 42 years' old

Gender: Male

Race: Black

First Language: isiXhosa

Agricultural Engineer

Has a wife and 2 kids

Uses Android device

About Thembile

- Constantly travelling around
- Has meetings daily
- Horrible sense of direction

Motivations

- His family
- BMW M5

Frustrations

- Traffic
- Loud noises

Scenario 2

As a visitor of the University of the Western Cape, Thembile finds himself facing the daunting task of having to park at the Life Science building and find his way to the Community Health Sciences building where he has to attend a meeting with the university rector and the dean of the science faculty. In addition, he happens to only have five minutes to find the meeting venue. Thembile often finds himself in this predicament when visiting UWC or any other university. Thembile would love to be able to easily navigate the university campus and reduce unnecessary stress before important meetings.

Hierarchical Task Analysis

Hierarchical Task Analysis for Student (Scenario 1)

- 0. To find venue/building
 - 1. Log in
 - 1.1. Enter student number
 - 1.2. Enter Password
 - 2. Get directions
 - 2.1. Enter current location
 - 2.2. Enter destination venue/building
 - 3. View Timetable
 - 3.1. Choose Specific Period
 - 3.2. Enter current location
 - 4. View Exam Timetable
 - 4.1. Choose Module
 - 4.2. View Test Information
 - 5. Venue Booking
 - 5.1. Query available venues
 - 5.2. Choose venue
 - 5.3. Enter time of use

Plan 0: If user is a student; you can do 1, 2, 3, 4, 5

Plan 1: If user is student; you can do 1.1, 1.2

Plan 2: If user is student; you can do 2.1, 2.2

Plan 3: If user is student; you can do 3.1, 3.2

Plan 4: If user is student; you can do 4.1, 4.2

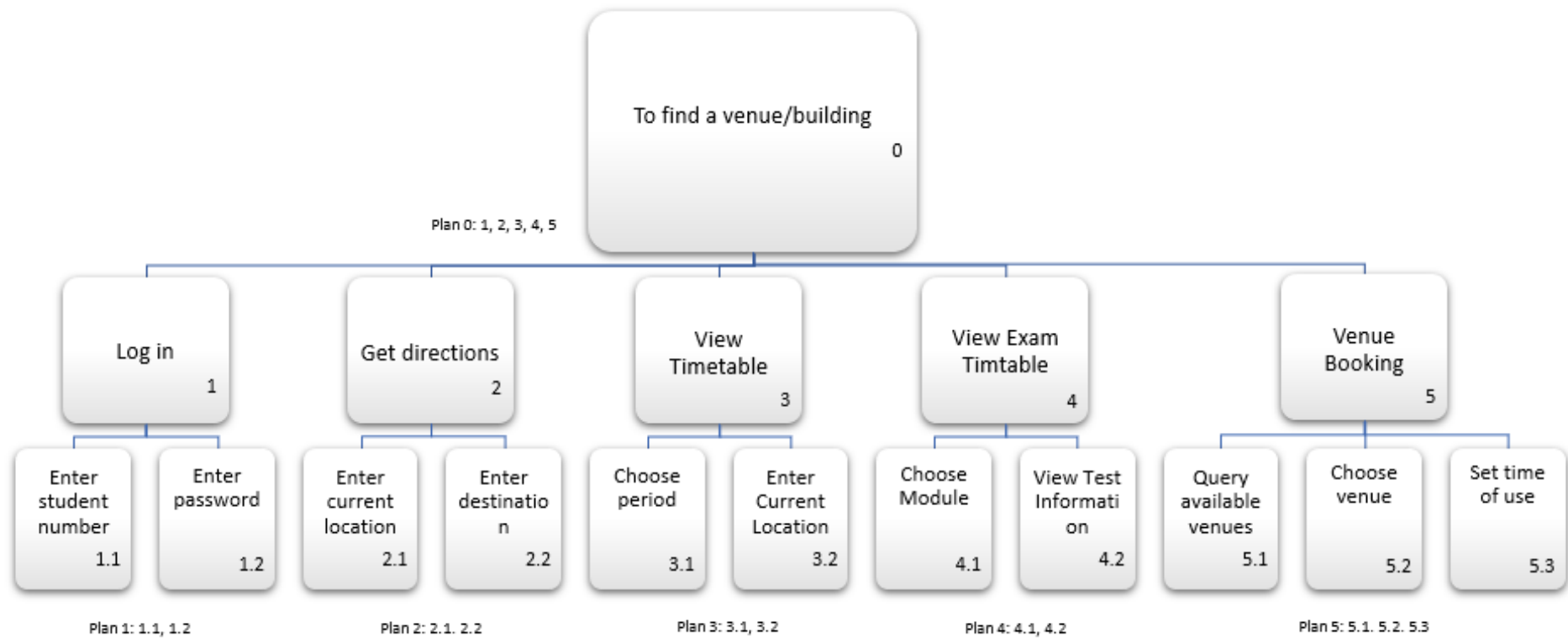
Plan 5: If user is student; you can do 5.1, 5.2, 5.3

Hierarchical Task Analysis for Visitor (Scenario 2)

- 0. To find venue/building
- 1. Log in
 - 1.1. Enter student number
 - 1.2. Enter Password
- 2. Get directions
 - 2.1. Enter current location
 - 2.2. Enter destination venue/building
- 3. View Timetable
 - 3.1. Choose Specific Period
 - 3.2. Enter current location
- 4. View Exam Timetable
 - 4.1. Choose Module
 - 4.2. View test information
- 5. Venue Booking
 - 5.1. Query available venues
 - 5.2. Choose venue
 - 5.3. Enter time of use

Plan 2: If user is a visitor; you can do 2.1, 2.2

Hierarchical Task Analysis Diagram



Requirements

Functional Requirements

- Once a student logs in their usage is tracked in order to provide future suggestions for recurring venues at a specific period on a specific day
- Note: A non-student user does not require logging in
- Once the user enters their current location:
 - Either through typing in their location or, grabbing their GPS location
 - Provide a path imprinted on the map to their destination
- Provide path through choosing period from timetable
- Provide venue booking facility

Data Requirements

- Student's login contains
{Student Number, Password}
- Users' getting a path contains
{GPS coordinates} – {Current location, Building, Venue}
- Student's timetable contains
{Venue, Day, Module, Period}
- Student booking venues
{Venue, Day, Module, Period}
- Student exam timetable contains

Environmental Requirements

Profile 1: Willem Steenkamp

Physical

Willem has a busy schedule during his days on campus. He has many classes to attend and as such, is constantly moving locations to get to the venues for his classes and society workshops. The UWC campus is large and venues are mostly scattered throughout the campus. There are always many people around campus walking to their different venues in a rush and the building locator signs on campus are few and not very detailed.

Social

Willem often arrives late to his classes and workshops due to the number of other classes and/or workshops he attends throughout the day. Since classes are scattered throughout campus and the time-period between classes is short, 10 minutes, he needs to be able to access the navigator app quickly with a clear interface and path to his next venue.

Organizational

Willem will be able to access the app for directional and venue-scheduling help at any time as long as he has a stable internet connection on his phone while he is on campus. Accessing the app off-campus doesn't help very much as its main purpose is on-campus navigation.

Technical

The app will be a mobile app. Willem is familiar with the use of a mobile phone. He is also familiar with the uses of mobile apps, and therefore won't find any problems using the Strut mobile app. He may not be able to access the full use of the navigator app if his mobile phone does not have a stable internet connection, as the app makes use of an internet connection for navigation.

Profile 2: Thembile Mafu

Physical

As a visitor of the University of the Western Cape, Thembile finds himself facing the task of having to park at the Life Science building and find his way to the Community

Health Sciences building. There isn't a lot of parking space at the Life Science building. The direction and view of the Community Health Sciences building isn't within sight from the

Life Science building. These locations are, however, not far apart and accessing parking at the Life Science building is very easy from the campus entrance near the Chemical Sciences building. It is not easy to find from the campus entrance near the Rugby Stadium as these locations are far apart and there are many roads on campus between these locations.

Social

Thembile happens to only have five minutes to find the meeting venue from the time he gets to campus and he gets lost easily. Thembile would love to be able to easily navigate the university campus and reduce unnecessary stress before important meetings. He needs he needs to be able to access the navigator app quickly with a clear interface and path to the Community Health Sciences building quickly and easily.

Organizational

Thembile will be able to access the app for directional help at any time as long as he has a stable internet connection on his phone while he is on campus. There are also security guards at the Life Sciences parking to aid in his direction. Accessing the app off-campus doesn't help very much as its main purpose is on-campus navigation.

Technical

The app will be a mobile app. Thembile is familiar with the use of a mobile phone. He is also familiar with the uses of mobile apps, and therefore won't find any problems using the Strut mobile app. He may not be able to access the full use of the navigator app if his mobile phone does not have a stable internet connection, as the app makes use of an internet connection for navigation.

User Requirements

User Profile 1

1. System providing on campus navigation
 - Building-to-Building
 - Venue-to-Venue
 - Building-to-Venue (and vice versa)
2. No learning burden on user
3. Functionality should reduce frustration and anxiety associated with navigation (Especially when lost)
4. Quick timetable access and path generation
5. Path generation based on different use-cases
 - Provide current venue as location
 - read location from GPS then
 - Set destination
 - Either from text entered or,
 - Read from timetable period selection

User profile 2

System providing on campus path navigation

- Current location-to-Building
- Building-to-Building
- Venue-to-Venue
- Building-to-Venue (and vice versa)

No learning burden on user

Quick access to search bar to search for destination

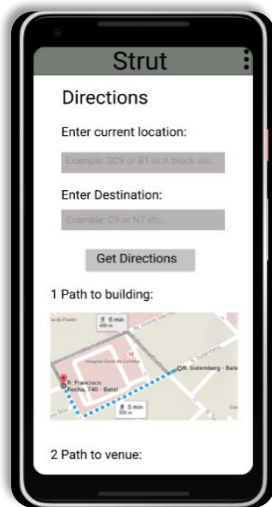
No learning burdens.

Usability Requirements

Willem is a fit individual that is a busy individual that is usually in a busy environment. He needs to be doing many different things during the day at specific times. He loves having a sense of accomplishment.

Thembile has a wife and 2 kids and is constantly traveling. His time is precious, and he is a very busy person. He also loves the bold design of the BMW M5 and is an agricultural engineer who values the good use of efficiency.

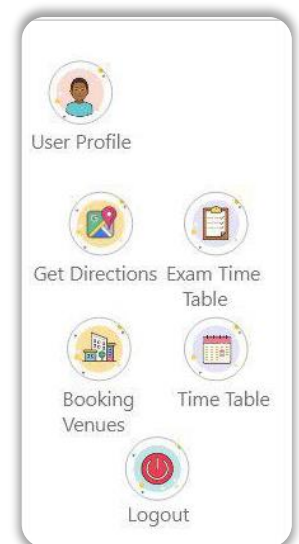
Conceptual Models



**Model 1: Model by Dean*

The [conceptual model](#) designed by Dean Ockhuizen implements a Map Metaphor and Timetable Metaphor. The model uses the Manipulating and Instructing interaction types to conduct activities and has a mobile interface.

The [conceptual model](#) designed by Tashreeq Waggie implements a Map Metaphor and Timetable Metaphor. The model uses the Manipulating and Instructing interaction types to conduct activities and has a mobile interface.



**Model 2: By Tashreeq*



***Model 3: By Tashreeq**

The [conceptual model](#) designed by Tashreeq Waggie implements a Map Metaphor and Timetable Metaphor. The model uses the Manipulating and Instructing interaction types to conduct activities and has a mobile interface.

Final Decision

We decided that Model 2 would be best suited for our app. The model was most applicable since it had the strengths from both model 1 and 3 incorporated. Model 2 best fits our interface metaphors and adds functionality for viewing exam timetables. Of the three it is also the most aesthetically pleasing and from this model we can build an app that reduces frustration and learning burden on the user's part through interactions with buttons.

Detailed Expansion

Interface Metaphors

The Map Metaphor evokes an initial mental model in users dating back to physical maps used to find directions before GPS.

The Timetable Metaphor depicts a softcopy approach to the timetable students receive on the day they register for class. This is especially useful in app since print timetables are large and require the user to constantly carry it around when starting out a new semester. This of course is solved by students being able to consult their cell phones (a device they always seem to have by their side).

Interaction Types and activities

Manipulating

Users explore the in-app map in order to find their way similar to real physical maps. This is used to generate paths for directions on campus between venues/buildings.

Instructing

Users explore the app by buttons to access the various options and enter location and destination information in order to get directions. Users also instruct app to query available venues for booking, timetable query, exam timetable query (Model 2) and login and logout.

Interface Type

Mobile - Handheld devices are portable and have become daily drivers in end users lives, especially in the lives of students. With this in mind, students will be able to use the app wherever they happen to be on campus.

Functions

- Allow students to login and logout
- Allow users to get directions from one venue/building to another venue/building
- Display paths to users (superimposed on map)
- Display in building directions to venue
- Allow students to display timetable and generate paths/directions to specified class
- Allow students to book venues
- *Allow students to view exam timetable (**Applicable to model 2**)

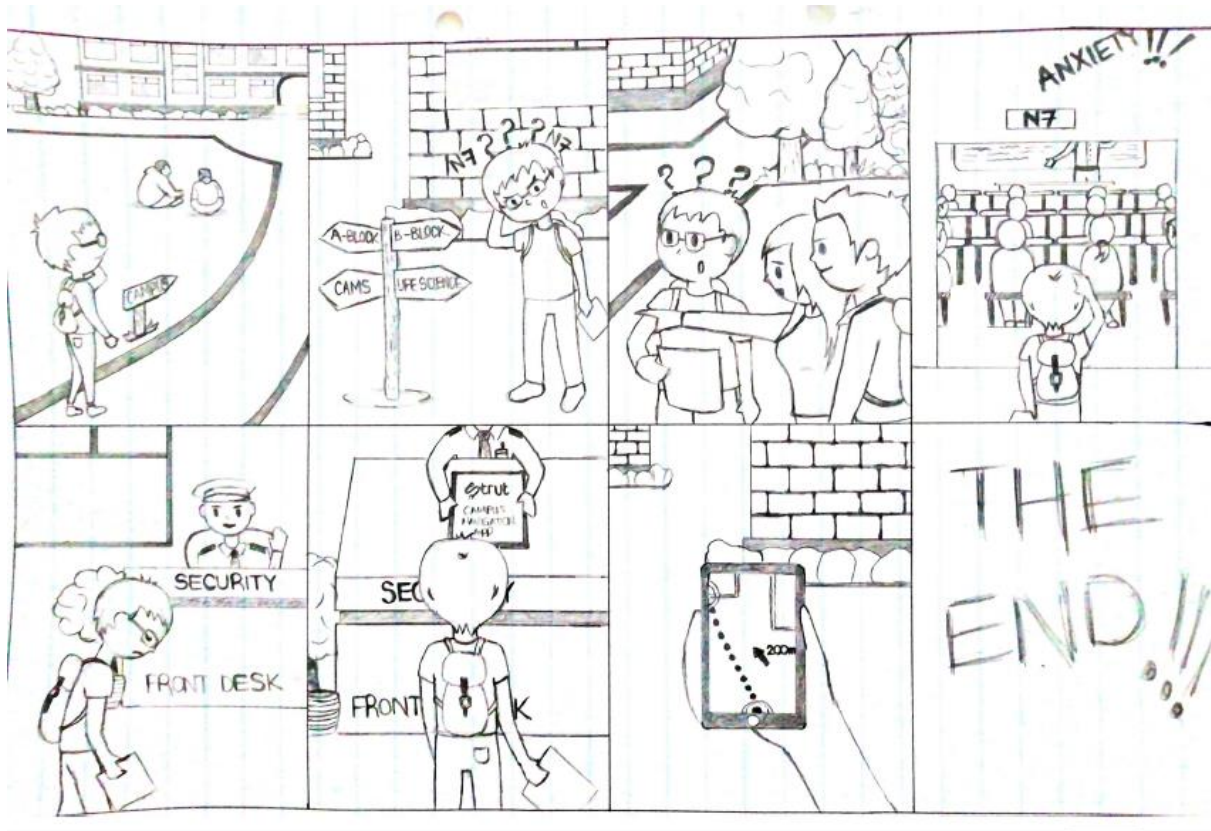
Function Relationships

- A student must login before being able to display his/her timetable
- A student must login before being able to book a venue
- A student must login before being able to display their exam timetable (**Applicable to model 2**)
- A user must enter an origin and destination before being able to display a path and map
- A path to a building must first be generated before being able to display in building directions

Information Requirements

- For login purposes
- Student Number
- Password
- In order to display a student's timetable, we need access to UWC's existing student database for module querying to build a timetable per student upon login
- In order to display the exam timetable, we will need access to the UWC's examination dates (**Model 2**)
- Bing Maps API for displaying locations and positions (buildings/venues on campus in our case)

Storyboard Prototype

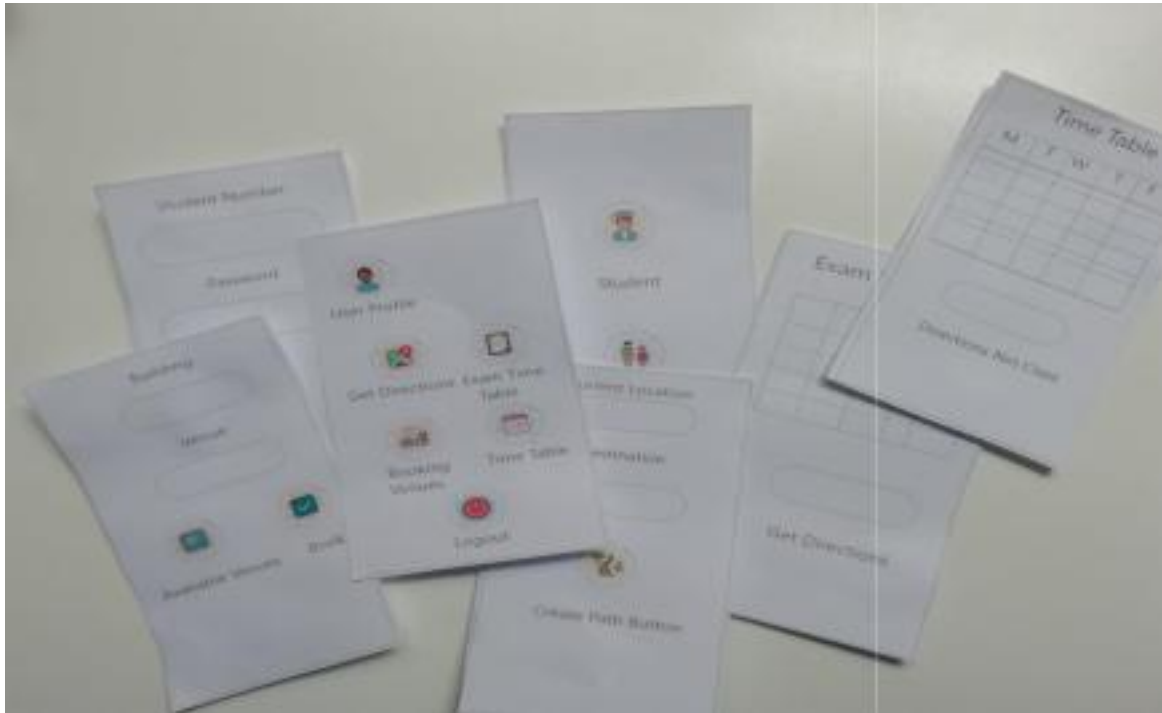


Feedback

We made users read our story and below are a few things the user saw could be better.

- User didn't know whether an individual that a student is not can make use of our mobile application as well. Create a screen that will allow user to be a visitor to just make use of the 'Get direction' screen.
- User ask how the student would make use of the app since every student has different degrees and modules so create a login screen for students using student number and their already made passwords.
- Timetable: We should make it visible that student would be able to get direction to class directly from the timetable screen by implementing a button.
- Should make it clear that student can get directions to exam venues from the exam venue screen by implementing a button.

Cardboard Prototype

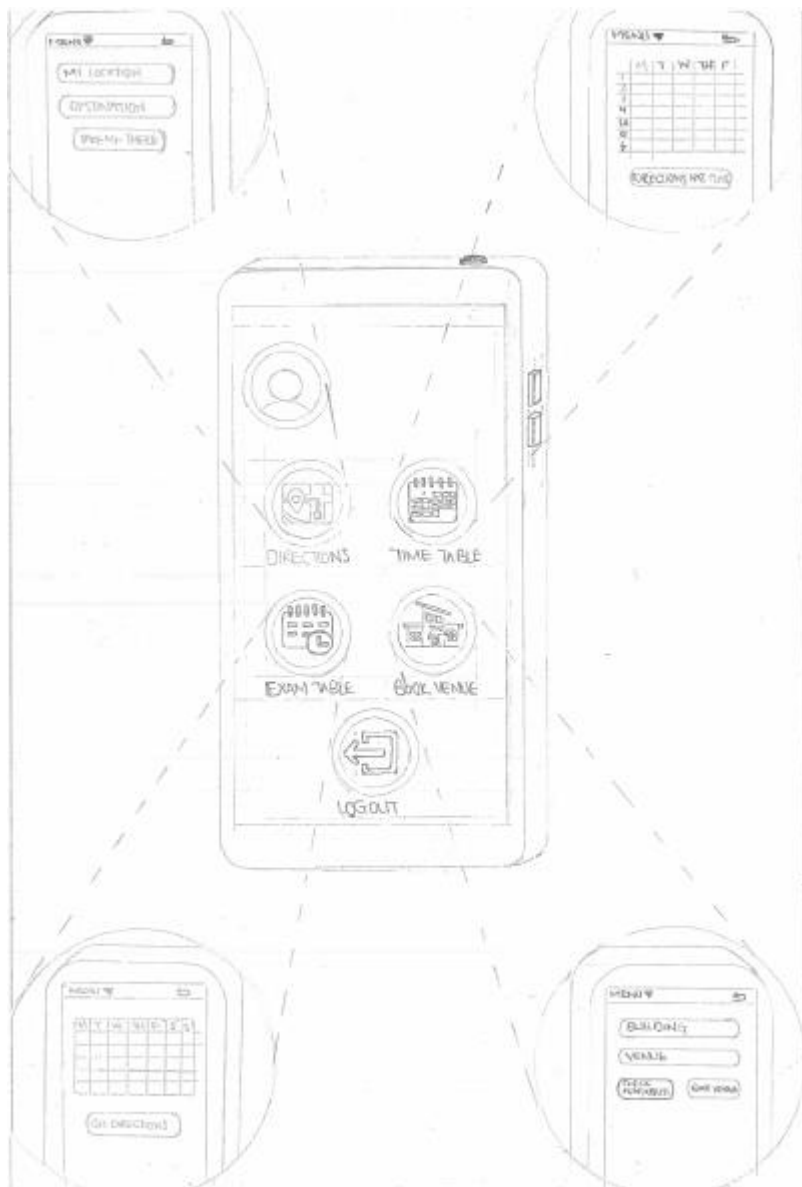


Feedback

For the feedback we had a user interact with the cards and perform tasks as if he was on actual mobile application. Below are a few things the user saw could be better.

- Back Button: The users made it immediately clear that we do not have a back button (Something we the creators of the application should have realized earlier).
- Timetable screen: User suggested why not just create a path to class when just tapping on a specific period.
- Get Direction screen: They would have like if a background of a map was adding, something similar to google maps for getting directions.
- The naming of buttons can be done better (grammar) and more to the point.
- Exam Timetable: Student should be able to get direction to an exam venue directly from the exam timetable screen by tapping on the module, making it convenient for users.

Sketch of applications main screen



Where am I?

This is the home (main) screen that the student will see after logging in using their credentials. The home screen will be one of the most interactive and allow for students to navigate through the mobile application.

What's here?

On the home screen there are 5 different buttons (which are icons). These buttons are labelled as followed – 'Directions', 'Time Table', 'Exam Table', 'Book Venue', 'Logout' and lastly a user profile in the top left corner. They will take users to their respective screens for functionality depending on which icon button was pressed. The layout is simple and pleasing. It's not cluttered, and users won't get

irritated with the home (main) screen. Lastly there is a user profile in the top left corner.

(You can read this if you want)

- Directions: An icon button labelled 'Directions' where the user clicks and takes the user to a different screen for getting paths to class. where they can enter their locations and destination(venue/building) that they would like to get a path to.
 - Timetable: An icon button labelled 'Timetable' where the user clicks and takes the user to a different screen to check their timetable. where they can view their timetable and get directions to a specific class
 - Exam table: An icon button labelled 'Exam Time' where the user clicks and takes the user to a different screen where they can see their exam table only related to their specific modules.
 - Book Venue: An icon button labelled 'Book Venue' where the user clicks and takes the user to a different screen to book venues.
-

Where can I go?

- The icon buttons give the user options to go to any other screens having their own function that fits together with the overall application
- Directions button will take the user to a screen where it will allow them to enter their current location and the destination location (such as the building or venue). You can press the button "Take me there" and it will generate a path on the map.
- Timetable button will take user to a screen where it will allow them to view their timetable, and there is a button 'Directions nxt class' which generate a path to the venue for their next period.
- Exam Table button will take the user to a screen where it will allow them to view their exam timetable specific to their modules they are doing. On the screen there is a button 'Get directions' that will generate path/directions for your latest exam or any one exam venue you would like know.
- Book Venue will take the user to a screen where they will be able to check which venues are available by entering the building (optional) and the

venue. Having then two buttons one for checking for availability and another to book then venue if its available.

- Lastly A logout button that will log the user out of their account and back to the login screen.

Is choice Usability or User experience

- Choice is for both usability and user experience.

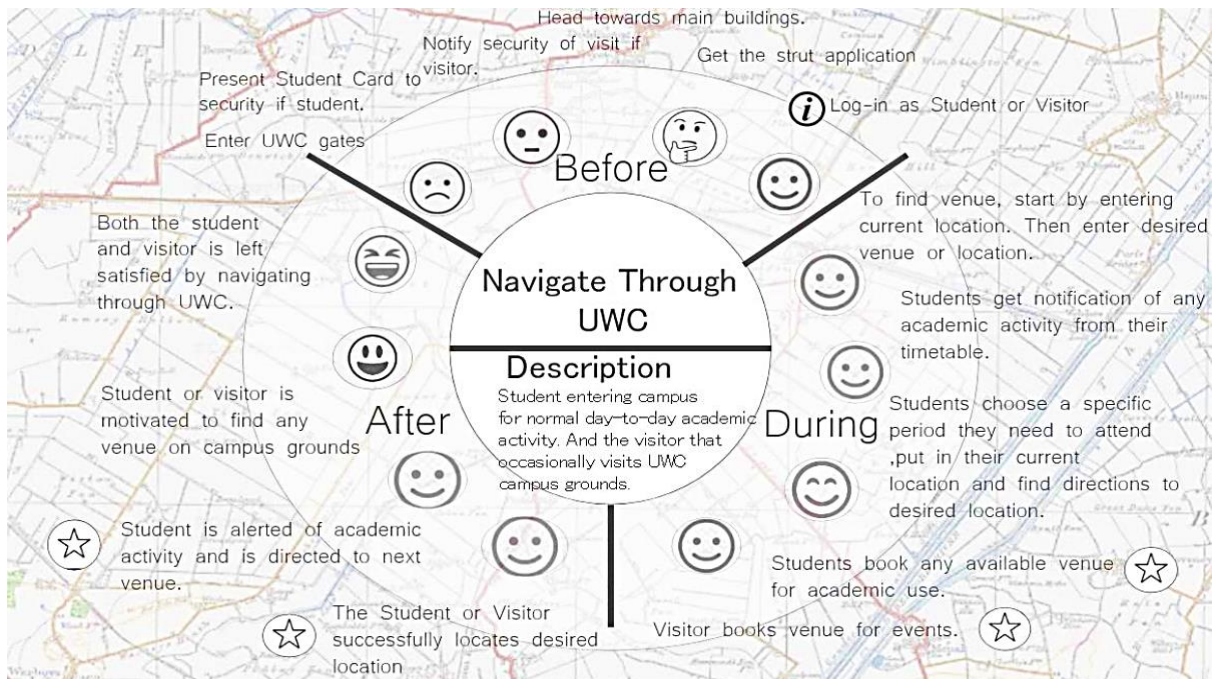
User experience

- It eliminates the stress of being late to lectures. One of the primary reasons why students at UWC are late to lectures is because they don't know where those venues are. Many a times students will fall behind in their course work because they have missed lectures because of being late. This results in stress and being demotivated. It will then affect one's health and marks later. This will lessen that burden and affect. It will hopefully help students in their academic lives and keep them on track.

Usability

- The system is laid out in the simplest possible format where everything that the students need to excess or may like to access is right in front of them. The main screens that are being used by students the most is not cluttered so they and straight to the point. We made it that so that its pleasing to look at an interact with. The buttons are icons which are more appealing to users of this day and age. When finding directions, the map shows paths as if you were looking at an atlas.

Experience map



i - data and information needed to deliver user experience.

😊 - The state of the user through the entire experience of using the Navigation application,

☆ - The functionality that will either satisfy the user or not.

New interaction issues

- Learnability
- Multiple holds: Not having the ability to reach buttons. This is what I mean:



How to address them

Multiple holds

Addressing the multiple holds issues, when designing the final interface, we can move the 'back' bottom or most of the icon buttons further to centre or closer to the bottom of the screen or create a gesture to replace it.

Learnability

We intend to create an information button on each page to indicate each page works, because in the case of Exam time table and the normal timetable users won't know that you can actual hold/tap on your specific period or exam venue detail and it allow you to get the directions for that venue. We all most likely take the user through a tutorial during their first use of the mobile application.

How does your app relate to the Maker Movement?

Our product is similar to the Maker Movement in that, the Maker Movement is characterised by individuals or a group of people collaborating on a project to create new and unique technology products. Our group too consists of a group of individuals collaborating together to create a new product which does not exists.

It differs in that, In the Maker Movement the people typically make use of many old, broken or discarded electronic components, whereas for our project, in terms of software we are essentially creating a new product from scratch by writing all the code and documentation by ourselves. However, it should be noted that we are using existing programming frameworks and languages.

What is the role of SDKs in this regard?

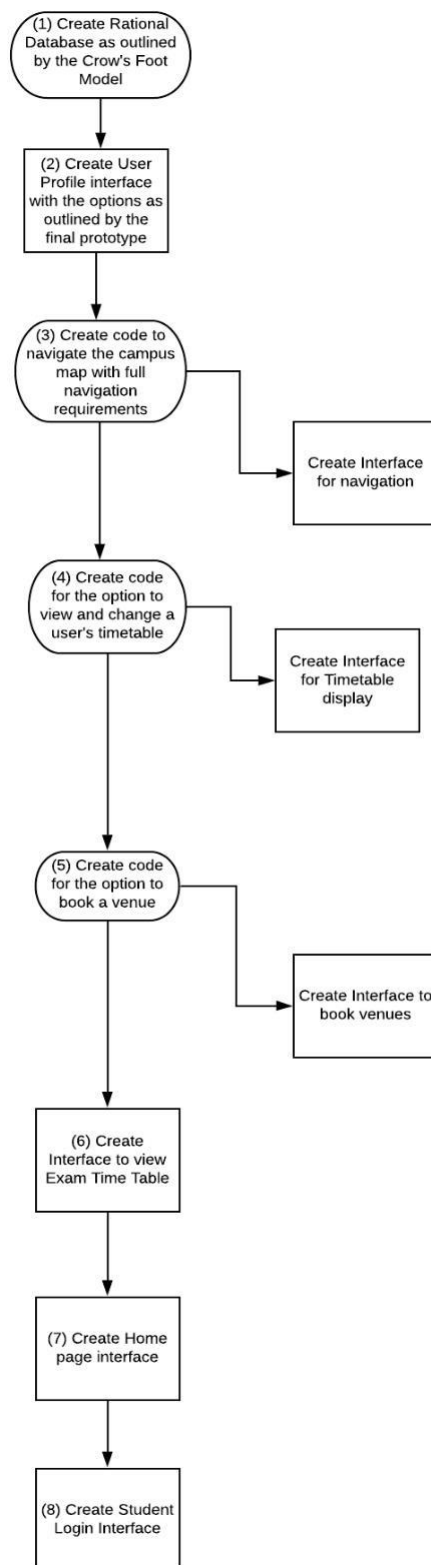
The role of SDKs is to facilitate the transition from designing to construction. The SDKs enable us to implement the aforementioned traits into our product because, they make the development process easier by including things like - Integrated development environment, documentation, drivers, and application programming interfaces (APIs) and more things which facilitate the construction of applications

Assuming AgileUX, what goes into Cycle 0?

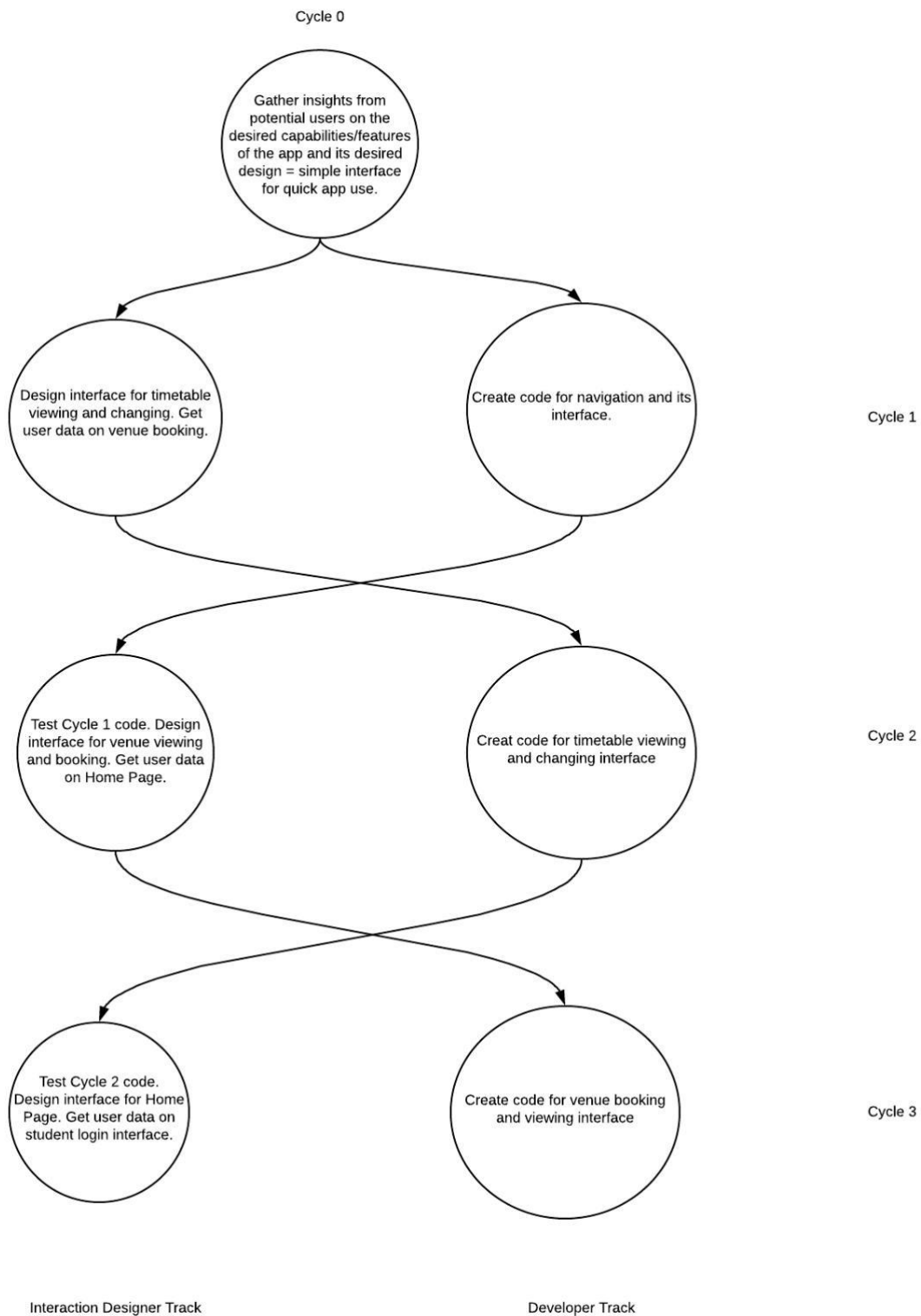
Cycle 0

The data collection from user profiles and the interviews with them has revealed that most potential users of the application have very little time to access and use the application. The application has to have a familiar and simple interface and produces detailed and non-complicated directions to the desired venues of the users as a primary function of the application. Secondary functions of the application need cater for the linking between student modules and their timetables. Please refer to the final prototype of the application for the overall design of the application.

Prioritise the requirements

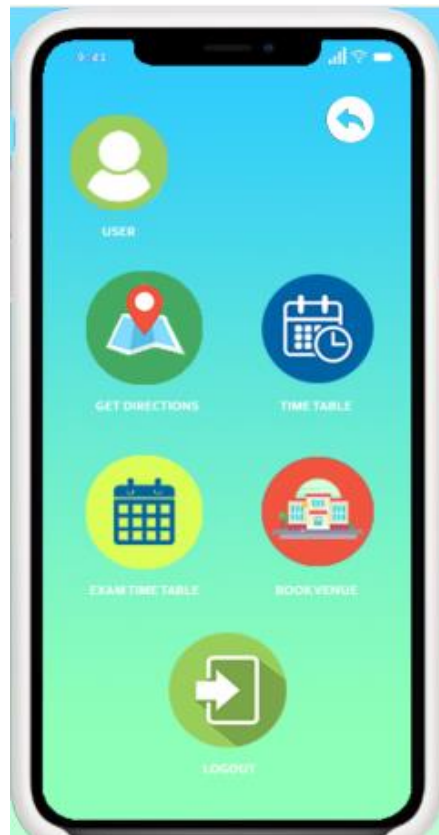


Sketch 4 cycles e.g. Cycle 0 and Cycles 1-3



Mock-up of landing page (Horizontal)

Link - <https://xd.adobe.com/spec/172e9885-670e-4168-5761-0f64ddba511a-611f/>



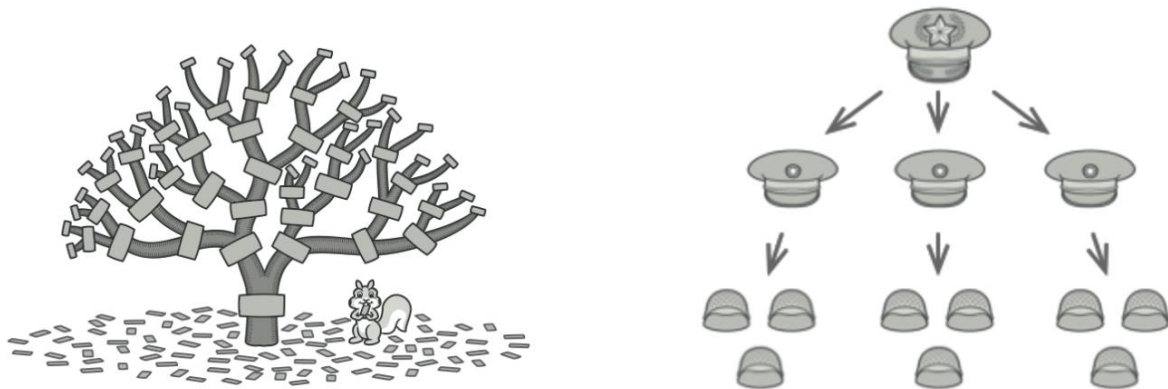
Prototype with Design Patterns

Link to final prototype with design patterns- -

<https://xd.adobe.com/spec/ee508e5f-3b0d-4a9c-61ab-25eb1d17b5e4-9861/>

Looking at our app there are two design patterns that come to mind which is Creational and Structural design patterns. In the case of Creational it will be our prototype that we are currently working on (above line). We are basically creating a copy of another application. We intend to let student find their venues on campus which is similar to google maps. In the case of structural, the app would be composite in that our main screen branches off into many other screens that has different functionality but to get to those screens you need to make use of the main screen. Just in the case of generating a path to find your venue you first have a screen that allows you to enter the details and takes you to another screen that displays the path

Some analogies for structural



Now looking at the prototype (vertical)



Capstone Assignment 3

Standard Task

Student

Directions

- You have never been to the CAMS building and would really love to try out the coffee at Vida Cafe. Find the path from your current location to your destination.

Timetable

- You are running late for your first lecture on a Monday morning. Check what module you have during period one on a Monday and then find the path from your current location to the lecture venue.

Exam Timetable

- You are taking the module CSC 312 and you have don't feel for the hassle of looking through an excel sheet for your exam venue and time. Find the time and place you are writing the exam and find the path to that exam venue.

Book Venue

- You are writing a test for CSC 312 in a few days and you want a place you can have group study session with your classmates. Check for available venues and book one.

Visitor

Directions

- You are delivering paper to the university of the western cape and have no idea where the library is. Login in as visitor and find the path from your current location to your destination.
- You have a meeting with the HOD of computer science but have never been to the CAMS building before. Login as visitor and find the path from your current location to your destination.

- You are wanting to make your way to the Admin building to pay your child's fees, but you have no clue where you are on campus. Login as visitor and find the path from your current location to your destination.
- You are running late attending your child's graduation and have no idea where the main hall is. Login as visitor and find the path from your current location to your destination.

Consent Form

Please read and sign this form

Strut is a mobile application in development. The purpose of the application is to help students locate lecture venues or exam venues easily without the students having gotten lost or asking for directions. Lessening their stress and anxiety.

The purpose of this research is to understand how real users interact with the product and improve the product based on the results.

In this research:

- You will be asked to perform specific tasks in the prototype on a smartphone.
- We will also conduct an interview with you regarding the tasks you performed.
- You will either be video, audio or written recorded.

Participation in this research is voluntary. You will not benefit directly from participating in this research. All information will remain strictly confidential. Your data will be password protected. The descriptions and findings may be used to help improve the Strut application. However, at no time will your name and any other identification be used. You can withdraw your consent to the experiment and stop participation at any time.

If you have any questions after today, please contact Tashreeq Waggie at

MTashreeqWaggie@gmail.com

I have read and understood the information on this form and had all of my questions answered

Date

Signature of participant

Name of Researcher
consent.

Date

Signature taking

User Interviews

Link to video and Consent form –

<https://drive.google.com/open?id=1SUA9yVcM3i82NQtMRAk36m8taAqD1qLP>

User 1

Test description

The test was conducted in the Sunlab in the CAMS building on the 9th of October 2019.

The user interacted with the prototype on an Android device.

Task

The user was requested to take on the role of a student.

The user was then asked to log in and thereafter navigate through the application.

Method

We explained the purpose of the application to the user and gave no further instruction.

The purpose of this was to gauge the usability and level of intuition the interface exhibits.

An interview was held post-experiment to understand why the user struggled with certain tasks and then enquire what changes would make the experience more friendly to the user.

Feedback

What I like

- The design of the system was simple and straightforward.
- The user was keen on having the sections of the timetable that are clickable resemble Buttons.

What I don't like

- The user clicked on the User icon on the home screen thinking it would display information not prompt them to log out.
- The user tried swiping on most screens because no return button was there or visible.

What I would change

- On the timetable view where users can click on venues to navigate there; the user did not know they could click on them. The user suggested a hyperlink style display to
- Let users know it is clickable.

Link to video and Consent form –

<https://drive.google.com/open?id=16yUYKHqJAiNy1YCdz01XQ8GglZp3m2Rq>

User 2

Test description

The test was conducted at my house on the afternoon of the 9th of October 2019.

The user interacted with the prototype via the browser on a laptop.

Task

The user used the application as a student would, to find venues at the university and possibly book venues or view their timetable.

Method

I explained to the user what the application is and its purpose. She then proceeded to use the application, while providing feedback at the same time.

Feedback

What I like:

The user liked the timetable functionality.

What I don't like:

The name was not clear and left users confused as to the name of the app.

What I would change:

The path displayed for the user to navigate to a building required more detail and needed start and end points to make it clear where a user is starting.

The map displaying the path should have more clarity.

The user suggests that we add the ability for a user to add notes to their exam Timetable.

The user did not like the colour scheme of the application, as she found it too colourful.

Link to video and Consent form –

https://drive.google.com/open?id=1VI5OZz_Y7zo2_IWdH499mNGiLLwdEQDj

User 3

Test description

The test was conducted on campus in the CAMS Building.

The user interacted with the prototype via the browser on a smartphone.

Task

The user used the application as a student would, to find venues at the university and possibly book venues or view their timetable.

Method

I explained to the user what the application is and its purpose. She then proceeded to use the application, while providing feedback at the same time.

Feedback

What I don't like:

- The written directions from and to a building were vague and were too simple. They need to be more precise.

What I would change:

- The application should allow the student to manipulate the timetable beyond academic activities. It should allow the user to change and insert activities at will.
- The application should help the user to navigate to and book for an appointment with medical assistance on campus grounds.
- The application should alert or notify user of any personal events.
- The application should allow students to record their marks.

Link to video and Consent form –

https://drive.google.com/drive/folders/1Vv226X_d9AncuX9DDI8wDTcl_8M0vZxv?usp=sharing

User 4

Test description

The test was conducted at the user's house. The user interacted with the prototype via the browser on a laptop.

Task

The user used the application as a student would, asked to carry out standard tasks provided by the researcher.

Method

I explained to the user what the application is and its purpose. I had the user complete certain standard tasks while screen recording to capture navigation and any errors. She then proceeded to use the application, while providing feedback at the same time.

Feedback

What I like

- Liked the colours and design of the app.

What I don't like

- Was confused with the functionality of some of the icons and the back button.

What I would change

- The user requested an initial tutorial window on the app for first time users.
- The user requested for the app to have the functionality of sharing user location with peers and generate paths to the each other.

Link to consent form

https://drive.google.com/drive/folders/1Ub_AKzidPTyMCN00M2-ZBYfpm3z5WtMT?usp=sharing

User 5

Test description

The test was conducted in the Sunlab in the CAMS building on the 9th of October 2019.

The user interacted with the prototype on an Android device.

Task

The user was requested to take on the role of a student.

The user was then asked to carry out standard tasks provided by the researcher.

Method

We explained the purpose of the application to the user and gave no further instruction.

The purpose of this was to gauge the usability and level of intuition the interface exhibits.

An interview was held post-experiment to understand why the user struggled with certain tasks and then enquire what changes would make the experience more friendly to the user.

Feedback

What I don't like

- I didn't like the start-up page, there's no indication if it's a loading page or if I have to click the icon to start.
- The exam timetable feature is confusing.

What should change

- I should be able to search for the module instead of looking for it on the timetable, unless only my subjects appear on the timetable then that's easier.

Field Study

In order to conduct our field study, we would have students use a mobile device (smartphone) pre-installed with Strut (our app) along with a screen recorder to measure the time it takes the student to understand the outline of all in-app functions.

We will have the students complete the following tasks

- Get directions to SC2 with departure point being CAMS
- In the timetable find the venue for CSC312 at 08:30am on Thursday
- From the timetable get directions from current location to MAT322 on Tuesday at 13:00pm
- In the exam timetable, find the venue for the final Mat322 examination
- Book Meeting Room 2 in CAMS for 09:40am on Tuesday if it is available

To avoid any bias being introduced into the study, we will refrain from recording the user's facial expressions as a form of data collection as this results in users changing their outlook and actions. Thus, losing focus of the tasks at hand which affects data collection. As such we will be recording voice, screen taps and time taken to complete each task along with an error count per task.

At the end of each separate student user study, the student will be given a short survey in order to query their frustrations or difficulties faced while completing in-app tasks.

The following needs to be accounted for:

- Students familiarity with the campus
- Students familiarity with Android/Apple iOS interface
- Students have different courses
- Students are from different year groupings (1st year, 2nd year, etc.)

The following types of data will be collected:

- Average time spent completing in-app tasks
- Time spent actually completing each task separately ([Refer to tasks above](#))
- Ease-of-use
- Data collected from the survey after engaging with the app

Data analysis:

- Line Graph used to reflect patterns in **Student vs average time for all tasks completed**
- Pie chart indicative of average time taken to complete each individual task (e.g. average time taken to get directions to venue across all students in field study)
- Statistics: Mean, Median, Mode for task taken longest to complete in field study
- Find mode of the tasks yielding highest error count
- Analyze screen recordings and identify patterns in student interaction with the app
- Using the information garnished from the surveys conduct qualitative analysis based on user satisfaction, frustration, what they liked, what they did not like and any changes the users might have suggested after completing tasks.

Controlled Study vs Field Study

Controlled Study

Benefits

- Controlled research is a research conducted in a setting specifically designed for research. This produces quantifiable data tightly linked to the participants' usage of the application.
- Due to the greater control that researchers are generally able to exert in a controlled setting than in a naturalistic setting, controlled research studies are more likely to represent a true experimental design of the application.
- Controlled studies have the advantage of greater control of irrelevant variables that might otherwise influence the results and thus possess clearer clues of the behaviours being observed when using the application. Within the context of the Strut application, the behaviours we were looking were how do the placements of certain buttons, icons used and the general layout of the pages of the app influence the efficiency of the navigation of the app by the participants. The data would be collected by video recording the participants using the app and their opinions on the degree of ease of use of the Strut app.
- This approach comes close to establishing a cause-and-effect relationship which increases understanding of the behaviours of the subjects when using the application. This can produce data which clearly highlights specific areas of performance of the application. This approach can determine the average time it takes a user of the Strut app to obtain directions between venues and the number of faults users made while using the app to obtain this primary objective of the app. This data establishes the cause-and-effect relationship between app use-time and fault occurrences to the app design.
- Another advantage of controlled research is its reproducibility. The environmental conditions can be neatly controlled and documented.

Problems

- The subjects in controlled research usually know that they are participating in a research study, which could bias their behaviours and hence the validity of the data linked to their observed behaviours and patterns when using the application.
- It may represent an artificial environment that may also influence the manner in which subjects behave and therefore alter results. This artificial environment does not simulate the UWC campus environment at all. Hence the results obtained in this idealistic/artificial environment will not produce accurate averages of the app use-time and fault occurrences when navigating the app.
- Maximizing the internal validity of the observed data linked to the subjects using the application requires very rigid control in a controlled setting, which in turn makes the research less realistic to a non-controlled setting. This implies that taking the subjects into an ideal environment when using the strut application doesn't take into account how typical environmental situations the targeted users of the Strut application will face influences their use of the application, and hence biases the data and the changes made to the Strut app based off the data. This bias implies that the app will be designed to cater only for the ideal environment and usage of the app, which is unrealistic.

Field/Natural Study

Benefits

- Field research is a research conducted in the real world or a natural setting. It tends to observe, analyse, and describe what exists rather than manipulating a factor under study. The research settings resemble the situations encountered in daily living, preserving the naturalness of the usage of the application. This field research took place at UWC and in-between lecture periods. This represented the true environmental situations users of the Strut app will face when typically needing to use the app.
- Participants in a field research may or may not know that they are being studied which reduces behavioural bias when using the application.

- The principal advantage of field research is its generalizability to real-life contexts because they represent a greater variety of situations and environments that subjects experience in their natural settings.

Problems

- The lack of control and the impossibility of precisely characterizing the UWC environment may make it very difficult to judge the generalizability of the observed behaviours of the participants when using the application and how it links to improvements to be made to the application.
- The more natural the experimental setting becomes, the more difficult it becomes to control extraneous variables.

Conclusion

The most fruitful overall research approach is usually to use both, controlled and field research. The results of controlled experiments produce new approaches or hypotheses to be tried and investigated in the field. Conversely, observations in the field produce new hypotheses to be tested by controlled experiments. This can be achieved by letting participants in the study video record their usage of the strut app while at UWC and in the situations they would find themselves in when needing to use the app, such as in-between lecture periods and venues. The researches would also be required to be in the field with them while they use the Strut app. We would need to install software on their phones to record their screens while using the app in their typical settings. This approach maintains the level of observation of the Controlled study while allowing the participants to be in the field and giving the researches a qualitative understanding of the data collected from this approach. This approach makes both data collection and analyses easy and accurate from both the quantitative and qualitative perspectives.

- Hassan A. Aziz, Archives of Clinical and Biomedical Research, Volume 1, Issue 2, 28 April 2017

Capstone Assignment 4

Heuristic Evaluation

Link to forms -

<https://drive.google.com/drive/folders/1kHc0kAScn9ipZepZLFP7zczV6m44WYCn?usp=sharing>

Consent Form

Strut is a mobile application in development. The purpose of the application is to help students locate lecture venues or exam venues easily without the students having gotten lost or asking for directions. Lessening their stress and anxiety.

The purpose of this research is to identify usability problems in the user interface design. Involving evaluators examining the interface and judging its compliance with heuristics

In this research:

- You the expert will perform the heuristic evaluation for our prototype/s for a standard task.
- You will evaluate our prototype and give feedback for each heuristic.
- You will be audio or written recorded.

Participation in this research is voluntary. The expert has been given all the information about using the prototype and the standard tasks that has to be performed. The feedback will be used to greatly help improve the Strut application. You will not benefit directly from this research. You can withdraw your consent to the experiment and stop participation at any time.

If you have any questions after today, please contact Tashreeq Waggie at

MTashreeqWaggie@gmail.com

I have read and understood the information on this form and had all of my questions answered

Date

Signature of participant

Name of Researcher

Date

Signature taking consent.

Briefing

Link to audio –

<https://drive.google.com/file/d/123ojXAe93U3bNnMej4Lbt2QdivUuR w-Y/view?usp=sharing>

(After this recording we decided to go with all the different methods under the heuristics. In this recording we said 9 but changed our mind to all instead.)

The reason we chose all is because we will be given feedback on a bit of everything so when we are coding our app, we can better it even in the smallest parts/or ways

Overview Process

5 experts were chosen from Group 2 to analyse and test the functionality and usability of the prototype. Having 5 experts allows us to get diverse feedback that will help us in making changes to the prototype.

Each of the experts were given a session on the prototype to familiarize themselves with the prototype. The second session was a brief on the standard tasks of the app, heuristics being used and was given unlimited time with prototype. The environment was not controlled as the interview was in the SUN lab, and even had the option of going to an isolated spot.

After testing the prototype, the feedback per heuristic of the experts is compiled, and any similarities are filtered out.

Heuristic Analysis Method: Jakob Nielsen (Sections 1-10)

A heuristic evaluation is a usability inspection method for computer software that helps to identify usability problems in the user interface (UI) design. It specifically involves evaluators examining the interface and judging its compliance with recognized usability principles (the "heuristics").

Evaluation Period

Compiled heuristic's evaluations

Visibility of system status.

What I liked:

There is a response to the click of a button as feedback to the user. When the user clicks in an area that is not clickable the app flickers the buttons on that page to notify the user of where to click, the system in this way notifies the user of their correct or incorrect input. After a click that app immediately moves to the screen as visual feedback of the user's input. The app displays dotted lines over the path that user is to follow from their location to their intended destination.

What I don't like:

There is a clickable area in the corners of the app which is a bit far from all the clickable areas on the screen.

What I would change:

All clickable buttons and icons would be moved to within reach of the user's fingers.

Match between system and real world.

What I liked:

The icon used in the app uses real world conventions such as map, timetable and building icons on the app's landing/ home screen. These icons bring the sense having familiar items in the system.

What I don't like:

The app uses system-like language instead of common English phrases. The button for finding a venue is labelled "get directions", which is language common to programmers.

What I would change:

The phrases used in the app, from their system-like commands to more common English phrases.

User control and freedom

What I liked:

The user has a choice to use the app as a visitor or student. If the user has mistakenly click on student, they have a back button to return to the home screen. If the user has made a mistake entering location information the user is able click on the return button to retype the information.

What I don't like:

Once on the home page the user is able to click on the logout page which completely takes the user out the app instead of going to the log in screen.

What I would change:

I would add a cancellation button to the screen that ask the user to enter their location and destination, which cancels the process of searching for the path and directions.

Consistency and standards

What I liked:

The application displays good consistency the colours, buttons and styling throughout the different screens. This maintains a similarity among the screens causing no confusion to the user.

Error prevention

What I liked:

In the case of navigating to the wrong screen the drop-down menu that is at the top right-hand corner allow users to directly navigate to their desired screen.

What I don't like:

Not every screen in the app that has a cancellation button or back button. Once the user has clicked on a cell in the timetable no cancellation button is there to undo the choice in case of mistakes.

What I would change:

Have an undo or cancel button place on screens that appear as a result of the users input.

Recognition and Recall

What I like:

The app is a native app, residing on the users' phones. Making recognizing the app easy. Its logo is also indicative of its context - navigation. This makes the initial activation of the users' recognition of the app easy and quick. The homepage has a high degree of external consistency in its design, making recall of the required actions to follow easy - selecting a profile-type to login via a button. Icons are very detailed and easily portray the function of the icon, making it second nature for the user to navigate through the app after their first experience with the app.

Flexibility and efficiency of use

What I don't like:

The app has limited options and simple and detailed prompts. Hence the app is designed to be used efficiently. Since there are limited options the app isn't flexible. What I would change: I would add a drop-down menu to make it easier for users.

Aesthetic and minimalist design

What I like:

The app has a very simple and hence minimalistic design with a gradient colour scheme. The app maintains a round-design of its GUI components and this consistency in-combination with the simplicity of the design What I don't like: The colours throughout the app is quite boring. All it is, is shades of blue and green. What I would change: I would like a more vibrant background and colours throughout the app. Make it nicer to look at.

Help users recognize, diagnose, and recover from errors

What I liked:

This is not incorporated into the prototype but the efficient, clear and rigidity of the design of the prototype minimizes the occurrences of errors.

What I don't like:

There is no pop-up list of available buildings/venues to choose from.

What I would change:

I would add this list to the 'Directions' page when clicking on the boxes to be filled in for venue entering.

Help and documentation

What I don't like:

There is no directions/ help info anywhere on your app.

What I would like:

I feel that the app needs some help documentation such as a button or description in that info can display on the screen to let users know about interactions that they are able to do which are not necessarily clear.

Debriefing

Link to Audio -

https://drive.google.com/open?id=1F8pRf6oZ5bU_i-0YdQOYPLZWPKAi7AUd

Redesign, what has changed and Why

Link Original Prototype - <https://xd.adobe.com/spec/ee508e5f-3b0d-4a9c-61ab-25eb1d17b5e4-9861/>

Link Redesign prototype – <https://xd.adobe.com/view/f8812077-f980-44fd-7169-11cc466511f2-8eb0/screen/33e0b027-7392-4626-ad78-0bca605a2f69/Launched-From-Icon-?fullscreen>

Looking back through our feedback of usability testing and heuristic evaluation we have come to the conclusion of a few problems that needs to be addressed and changed for from the original prototype.

Here they are

1. In the redesign of the prototype we added a help documentation on the bottom of the timetable and exam timetable screens.

It was made clear that users did not know in both testing and evaluation that the Timetable screen and Exam timetable screens are interactive and that directions can be generated from these two screens.

2. In the redesign of the prototype we added a drop-down menu in the top right corner and scrapped the back button.

It was created because users were irritated and made it clear that having to go back to the main screen to just get to another screen was annoying and it wasn't making our application really flexible and efficient to use.

3. The logo has become a home button.

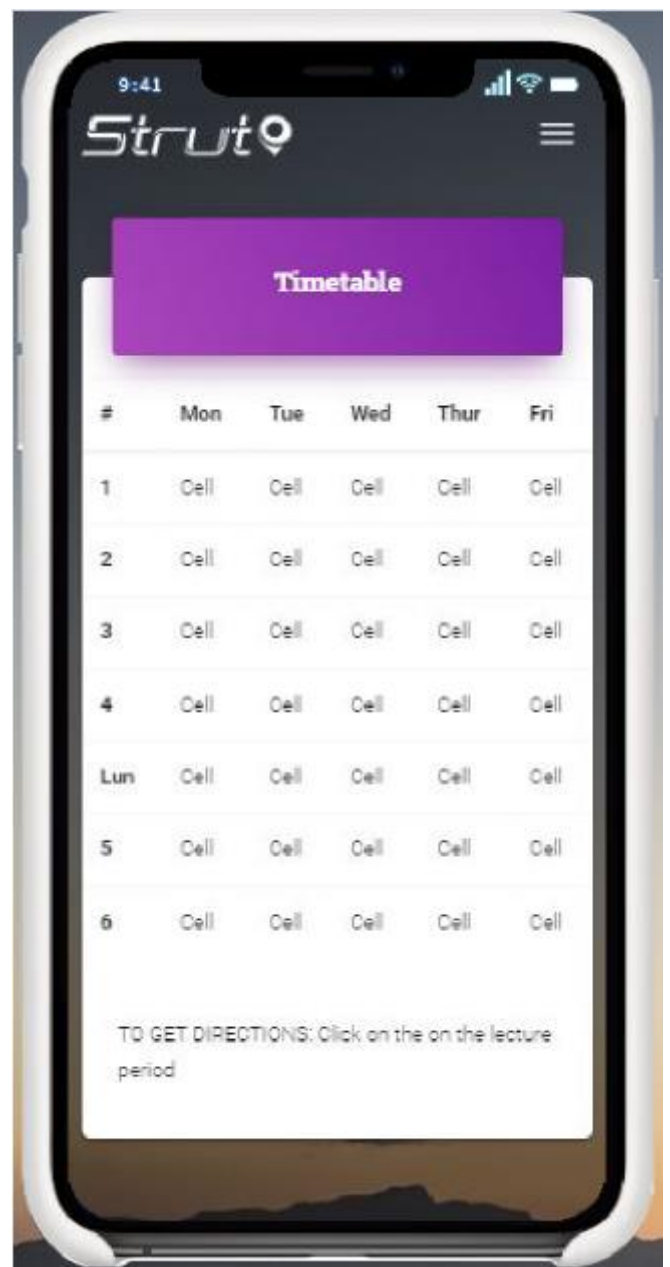
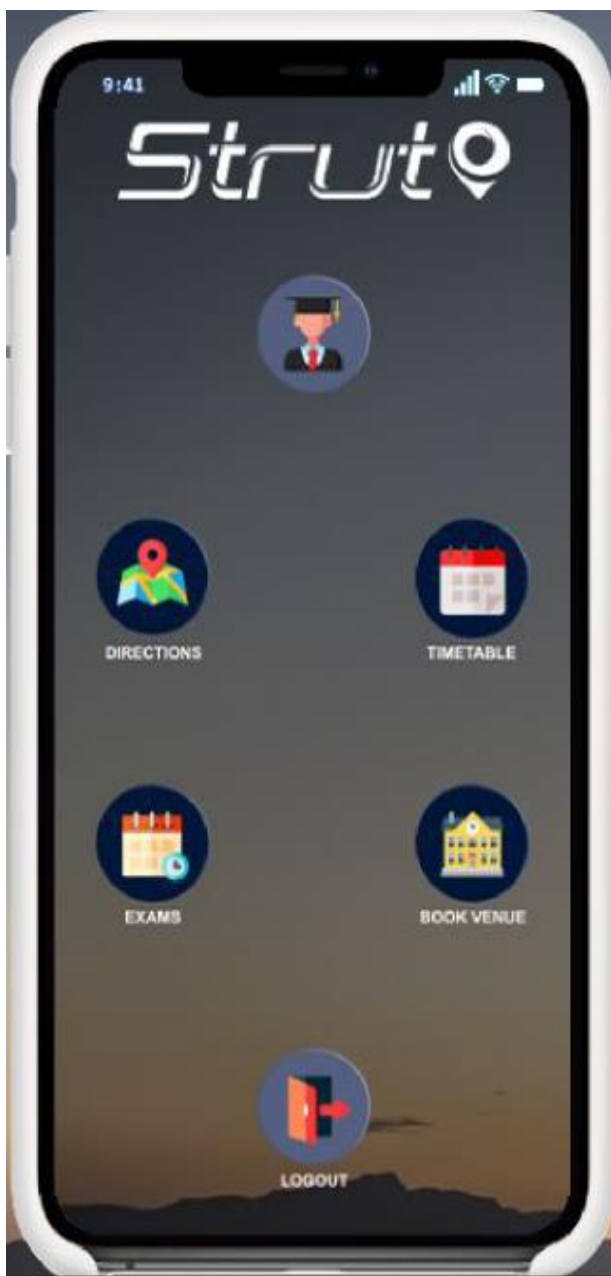
This was done to accommodate the drop-down menu and that if the user our like to go back to the home screen they can do it using the logo.

4. In the redesign of the prototype we changed the entire colour scheme

and background our Strut application.

This was done because users found the original scheme and background really dull. No one really wants to use a mobile application that isn't appealing to the eyes (Ikamva for an example).

Some Samples



Comparison to Previous Test

Findings in the previous tests

Most common errors (Usability Testing)

- Prototype was not interactive enough
- Users were not able to get directions from the Timetable and Exam Timetable screens.
- Couldn't enter details (Since it's a prototype we can't fix this)

Testing Stats (Usability Testing)

- Average error per user: 3.66
- Average test time per user: 4 minutes 38 Seconds

Identified problems (Heuristic Evaluation)

- There is no help documentation to help users
- The overall interface isn't appealing
- Flexibility and efficiency of the Strut application is limited
- There are no error message indicators. (Because the limitation of the prototype software we aren't really able to incorporate the way we want)

Findings and differences observed

It was clear from previous tests that users would encounter errors and that those errors should be looked at. In order for users to make use of the app, the app should be interactive, and users should be provided with the correct heuristics to prevent them from getting "stuck". These findings are important in the prototyping phase so that it prevents developers from fixing errors after the deployment phase.

Differences after fixing errors and making changes to the heuristics

- User could navigate better (Drop down menu).
- Interaction was still limited but not like before (but it's still a prototype).
- Users were now able to tell if a page had additionally interaction that couldn't be visible before.
- Amount of errors and time taken became less.

The second time around the app is usable, the users must be able to perform whatever task is required. In the previous test's users would get stuck on pages and not know what functionality was available to them. After fixing the errors, users are able to perform tasks without being stuck or having major errors.

What differences do you observe (in terms of evaluation approach)?

I observed that with heuristic evaluation, experts/users look at the user interface and identifies the problems. The usability testing on the other hand, potential users try out the user interface with real tasks (The standard tasks we created). The problems found with usability testing are true problems in the sense that at least one user encountered each problem, such as users not being able to get directions from the Timetable and Exam Timetable screens. The problems found with heuristic evaluation are potential problems – the evaluator suspects that something may be a problem to users such as users becoming irritated with the fact that they must go back to the main screen just to go to another functionality screen.

Which evaluation approach do you prefer and why?

Our team prefers the usability testing approach and the reasons for that is, it allows our team to get direct feedback from the target audience. It highlights issues and potential problems before the launch of the product. It minimises the risk of the product failing.

In conclusion usability can be used in a variety of ways. Despite the fact we couldn't mimic real life usage, we found usability testing to be the best method to ensure users achieve goals quick and easily.

Comparison Table

	Usability Testing	Field Studies	Heuristic Evaluation
Findings	<ul style="list-style-type: none"> During the usability test, users were asked to perform specific tasks. We found that most of the errors resulted from the tasks of finding directions from the timetable and exam timetable. This could be that the user is inexperienced with the app or we have not properly conveyed the interactions. 	<ul style="list-style-type: none"> We found that users had more time to play around with the prototype and have a better feel for it. They made less errors than those users of the usability test because they could still play around with the prototype and get a feel for it. 	<ul style="list-style-type: none"> We found that often the problems that were being identified were not critical. It more had to do with the efficiency and flexibility of the interface itself. Example that it wasn't really efficient or flexible that the user had to keep going back to the main menu to user a different functionality
Benefits	<ul style="list-style-type: none"> You learn how long it takes a user to complete any of the standard task and how much errors per standard tasks they have made. For example, getting the directions to a building such as CAMs. This also allows you to find out whether you are designing for a good user experience and how satisfied users are with a particular prototype. 	<ul style="list-style-type: none"> The benefit is that user takes the prototype into the real world and experience what it is like. This is good because it gives you feedback on how to shape and change your final product that it will be used and enjoyed by others in the real world. We did this in assignment three where we let one of the users make use of the prototype in their own time without our assistance. You can send the prototype to anyone, you don't need much resources. 	<ul style="list-style-type: none"> Since the evaluation is done by several people there is a better chance of getting a better range of views and identified problems.
Costs	<ul style="list-style-type: none"> Makes use of a lot of resources and takes a lot of time to set up. For example drawing up consent forms and, finding a suitable environment/place to host the testing and finding users that are free and willing to participate 	<ul style="list-style-type: none"> When users are using your prototype, you have no control over them. During assignment 3 we gave sent the prototype to users and they got back to us with feedback. You might not always get feedback right away unlike a usability test. 	<ul style="list-style-type: none"> Experts are required and this can be time consuming and expensive to research and set up. The more experts to do the evaluation the better. This results in having to spend a lot of time analysing and reviewing experts to make sure they are experienced in the issues you are concerned with regarding your application
Limitations	<ul style="list-style-type: none"> Time is not always suitable, Students/Candidates are not always available, they might not be at campus or have class to attend 	<ul style="list-style-type: none"> Users cannot be recorded Only can collect limited data 	<ul style="list-style-type: none"> Experts may be difficult to find. In our case we were only able to get a hold of 5 members of group 2 because of time The experts might/are costly.

Software Engineering

Process Model

During Software Engineering we have learned a few different process models and the one our group settle on was the Agile (Scrum) process model. The reason we chose this methodology was because it is based on teamwork, it allows for close collaboration with customers and stakeholders, flexibility and ability to adapt to change in requirements and better manage all tasks. It allows use to iteratively gather requirements throughout the development process. The building blocks of Agile are planning, analysis, design, development and testing. The idea is to produce documentation with information to move forward when it makes sense.

General Principles

- Abstraction

The user will only be shown digestible information rather than technical details which is hard to understand.

- Localization

The app will only display to the user information which is relevant to their current location query.

- Hiding

The user will not be shown the background workings of the application, they will only be presented with the results of the background activities which will help navigate them towards their destination.

- Completeness

The application will fulfil all the user requirements, which in this case is providing sufficient information to users so that they may reach their requested destination.

Goals

- Understandability

The application will be easy to use, and users will be able to quickly identify how to use the application. When users use the application for the first time, we plan to construct and implement a walkthrough which will take a user through steps of how to use the application.

- Reliability

The application will be reliable to use without losing any of its functionality or crashing at random times leaving users unable to use it.

- Security

Users who are not visitors will always need to be logged in with their own login details in order to utilize the application so that other users may not have access to their details and information. The data we store in the database will not be available to unauthorized persons, this is so that user's data that we store remains safe and unavailable to unauthorized and/or third parties with malicious intent.

Umbrella Activities

- Software project tracking and control

Tracking the progression of the project is key given the short amount of time at hand to complete tedious and complex tasks. Our team is using Agile Scrum Methodology which best suits for the short time and uncertain timetables of our team members. Having daily sprints of various time frames will allow the team to keep track of accomplished tasks to date.

- Software Quality Assurance

Making sure the software meets the functional requirements of the project. Given that the system is to work on a daily basis, careful consideration will be put into building the application to cope with daily operation. The application will also be built with familiarity to presently operating navigating systems to ensure that the system is usable.

- Technical Reviews

Examining the suitability of the application for daily navigational use and identifying any discrepancies from specifications and standards. Reviews gathered from testing. Testers will include Developer, Students and Visitors.

Technical Reviews will be gathered from student's initial full interaction with the application, to iron out any inconsistency that may occur.

- Software Configuration Management

Software Configuration will be occurring every 2 months to update the paths recorded after the use of the app for those 2 months by students and visitors that will be using the application. Updates to the paths will include new shortest paths to and from buildings, and possible new spots that students and visitors go to frequently.

Student timetables will need to be part of the software configuration. This accommodates for new modules that students will need to do in new terms, semesters or years. To support the app's exam notification functionality, the data from student's modules will be used to draw up an exam timetable personalized exactly for the student.

- Reusability Management

Some of the functionality of the software was reused and was made possible via Google's Map API's, since google maps is able to locate paths and paths to buildings.

Module data from students from a certain year with similar or same modules as newer students will be used to now personalize new student's academic timetables.

Engineering Principles that guide Process

- Be agile

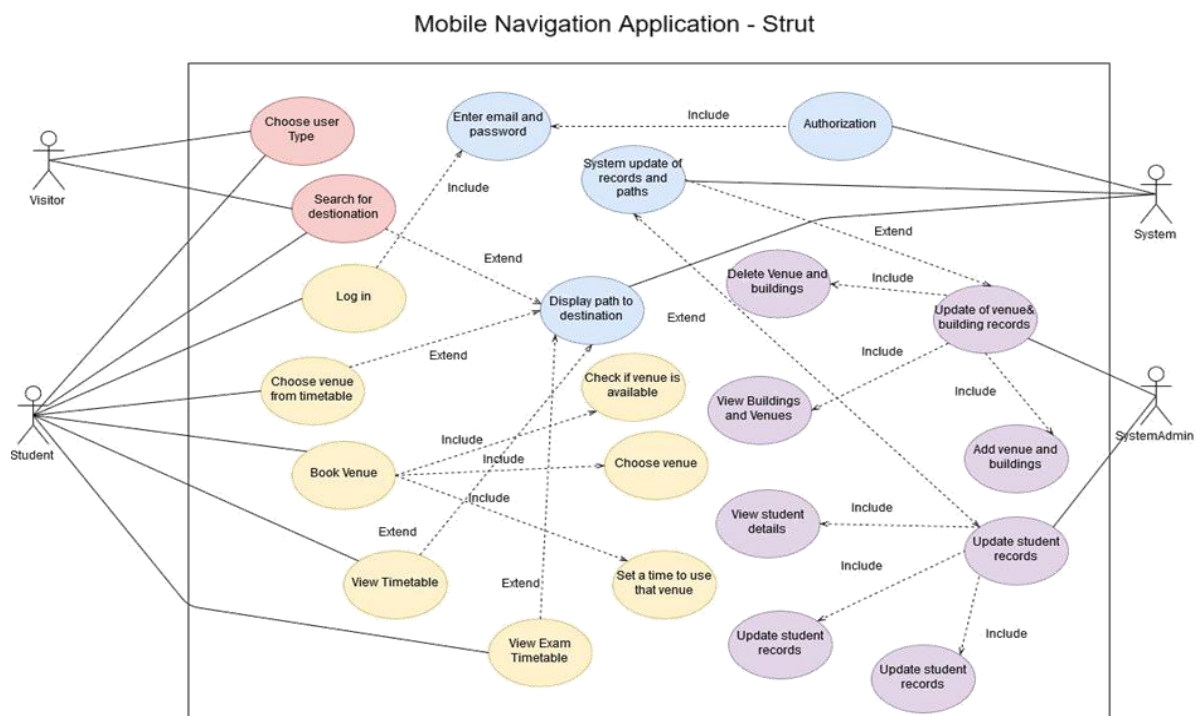
The process model we are applying is agile based, thus we can incorporate this principle into our process. Agile allows us as a team to get feedback from users about our prototype every week so that we can apply those changes and use that feedback to the product in development.

Software Engineering Principles that guide Practice

- Divide and conquer

Our App is divided into many different functionalities which has the team spilt into a dynamic that each part of the App is always being worked on and eventually bringing all of that together.

UML Diagram



Databases

Entities

- Student
- Enrolment
- Module
- Timetable
- Venue
- Building
- ExamTimetable
- BookedVenue

Business Rules

1 Student has M Enrolments; possibly none

1 Enrolment belongs to 1 student; not none

1 Module has M Enrolments; possibly none

1 Enrolment belongs to 1 Module; not none

1 Module is found in M Timetables; possibly none

1 Timetable contains 1 Module; not none

1 Venue is found in M Timetables; possibly none

1 Timetable contains 1 Venue; not none

1 Module is found in M Timetables; possibly none

1 Timetable contains 1 Module; not none

1 Venue is found in M ExamTimetables; possibly none

1 ExamTimetable contains 1 Venue; not none

1 Building has M Venue; possibly none

1 Venue belongs to 1 Building; not none

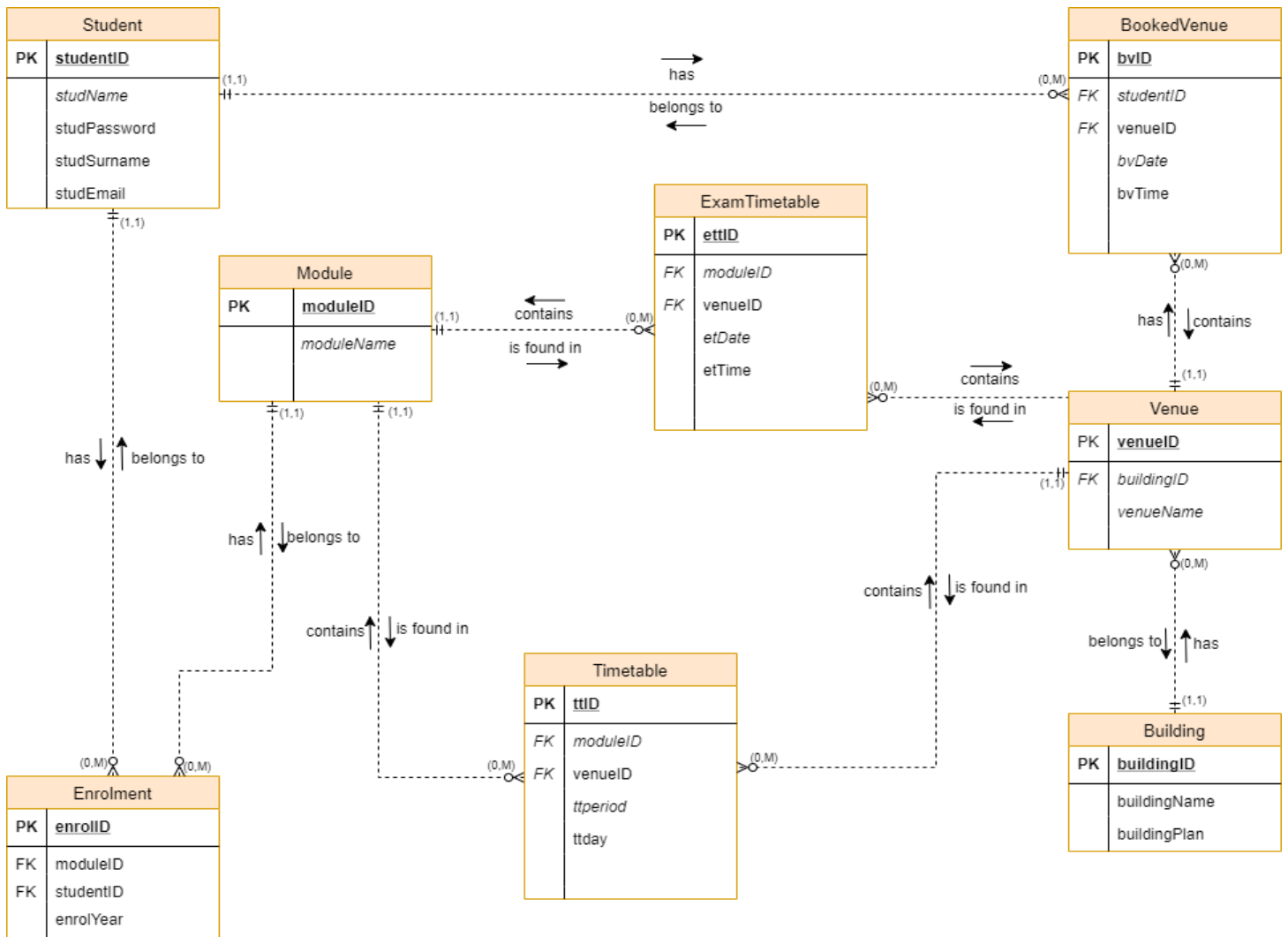
1 Student has M BookedVenues; possibly none

1 BookedVenue belongs to 1 Student; not none

1 Venue has M BookedVenues; possibly none

1 BookedVenue contains 1 Venue; not none

ERD Diagram



Internal model in MySQL, data types for attributes

```
CREATE DATABASE strutDB;
```

```
USE strutDB;
```

```
CREATE TABLE Student (  
    studentID int NOT NULL AUTO_INCREMENT,  
    studName varchar(255),  
    studSurname varchar(255),  
    studPassword varchar(255),  
    studEmail varchar(255),  
    PRIMARY KEY (studentID)  
);
```

```
CREATE TABLE Building (  
    buildingID int NOT NULL AUTO_INCREMENT,  
    buildingName varchar(255),  
    buildingPlan varchar(255),  
    PRIMARY KEY (buildingID)  
);
```

```
CREATE TABLE Venue (  
    venueID int NOT NULL AUTO_INCREMENT,  
    venueName varchar(255),  
    buildingID int,  
    PRIMARY KEY(venueID),  
    FOREIGN KEY(buildingID) REFERENCES Building(buildingID)  
);
```

```
CREATE TABLE Module (  
    moduleID int NOT NULL AUTO_INCREMENT,  
    moduleName varchar(225),  
    PRIMARY KEY (moduleID)  
);
```



```
CREATE TABLE Enrolment (  
    enrolID int NOT NULL AUTO_INCREMENT,  
    moduleID int,  
    studentID int,  
    enrolYear YEAR,  
    PRIMARY KEY(enrolID),  
    FOREIGN KEY(studentID) REFERENCES Student(studentID),  
    FOREIGN KEY(moduleID) REFERENCES Module(moduleID)  
);
```

```
CREATE TABLE Timetable (  
    ttID int NOT NULL AUTO_INCREMENT,  
    moduleID int,  
    venueID int,  
    ttperiod int,  
    ttday int,  
    PRIMARY KEY(ttID),  
    FOREIGN KEY(moduleID) REFERENCES Module(moduleID),  
    FOREIGN KEY(venueID) REFERENCES Venue(venueID),  
    UNIQUE KEY UC_Timetable(venueID, ttperiod, ttday)  
);
```

```
CREATE TABLE ExamTimetable (  
    etID int NOT NULL AUTO_INCREMENT,  
    moduleID int,  
    venueID int,  
    etDate DATE,  
    etTime int,  
    PRIMARY KEY(etID),  
    FOREIGN KEY(moduleID) REFERENCES Module(moduleID),  
    FOREIGN KEY(venueID) REFERENCES Venue(venueID),  
    UNIQUE KEY UC_Timetable (venueID, etDate, etTime)  
);
```

```
CREATE TABLE BookedVenue (  
    bvID int NOT NULL AUTO_INCREMENT,  
    studentID int,  
    venueID int,  
    bvDate DATE,  
    bvTime int,  
    PRIMARY KEY(bvID),  
    FOREIGN KEY(studentID) REFERENCES Student(studentID),  
    FOREIGN KEY(venueID) REFERENCES Venue(venueID)  
);
```

Appendix

Appendix A

NAVIGATION APPLICATION INTERVIEW QUESTIONS

NB ! - Explain the project and ask for consent

Do you make use of Android or IOS smartphone?

Are you or were you a university student?

If the individual has replied 'Yes'

What are the buildings you make use of the most on campus? (If they were or are a student of the University of the Western Cape)

Have you ever missed or been late to a lecture, practical or tutorial due to being unable to find the desired venue?

If they respond 'yes' to the previous question, have you ever just felt that it's a waste to go now that you late? Does it demotivate you in anyway?

How often do you get lost while looking for your desired venues?

How often do you ask for directions from people around on getting to your desired location?

Have you ever arrived late to an exam venue?

If they respond 'yes' to the previous question, has arriving late affect your mental or emotional state when writing and how?

During your time at university have people given you incorrect instructions to finding a venue

Did you know that University of the Western Cape has a map of campus showing all the venues on their website? (If they were or are a student of the University of the Western Cape)

Do you prefer the current way to find your lecture or exam venues?

Why?

Would you benefit from an application which could provide you with a direct path from your current location to your desired venue?

Do you think other students, staff, lecturers or visitors could also benefit from such an application?

What would make you regularly use such an application in terms of interface

What would not make you use a navigation application to find your venues

If the individual has replied 'No' to the first question

Have you ever visited a university or college? (Any tertiary institution)

Do you get lost while looking for desired building?

Do you ask for directions from people around you on getting to your desired location?

Would you use a mobile application to find the location you are looking for even if it is just once off?

Appendix B



Model 1. Model by Dean Robinson

Appendix C

Student Number

Password

User Profile

Get Directions

Exam Time Table

Booking Venues

Time Table

Logout

Building

Venue

Available Venues

Book

Current Location

Destination

Create Path Button

Time Table

M	T	W	T	F

Exam Time Table

Appendix D

Login

Student Number

3537968

Password

Login

Register

Student Number

3537968

Password

Main Options

Map

Get Directions

Book Venue

Timetable

About

Logout

Book Venues

All available venues in a buidling

A Block

A1 Book

A2 Book

Get Directions

Enter Location

B1

Enter Destination

Sun Lab

Path to building

Path to Venue

Go

Time Table

	M	T	W	T
1	CSC	MAM	IFS	MAM
2	IFS	CSC	CSC	

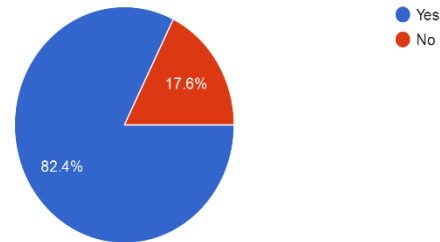
Click on period and hold

Choose Period and Enter current location

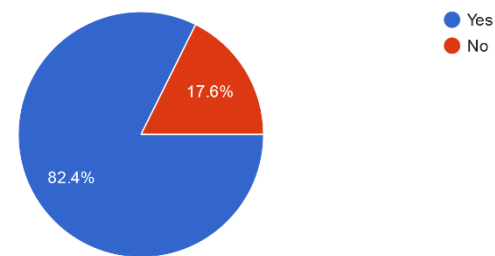
Get Directions

Appendix E

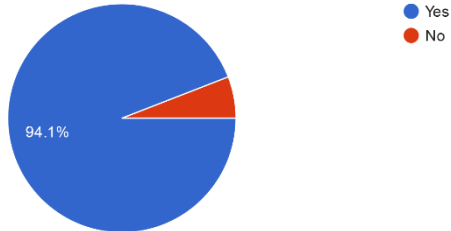
Have you ever missed or been late to a lecture due to being unable to find the desired venue?
34 responses



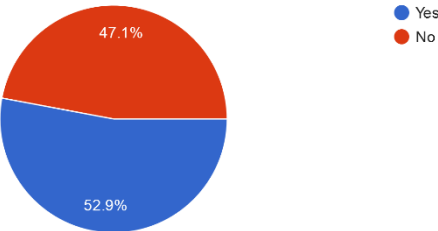
Have you ever gotten lost while looking for your desired buildings or venues on campus?
34 responses



Have you ever asked for directions to get to your desired building or venue?
34 responses



Has arriving late to an exam affected your mental or emotional state while writing?
34 responses



Appendix F

1	Visibility of system status <ul style="list-style-type: none">• The system should always keep users informed about what is going on, through appropriate feedback within reasonable time
2	Match between system and the real world <ul style="list-style-type: none">• The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3	User control and freedom <ul style="list-style-type: none">• Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo
4	Consistency and standards <ul style="list-style-type: none">• Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5	Error prevention <ul style="list-style-type: none">• Even better than good error messages are a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6	Recognition rather than recall <ul style="list-style-type: none">• Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate
7	Flexibility and efficiency of use <ul style="list-style-type: none">• Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8	Aesthetic and minimalist design <ul style="list-style-type: none">• Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility
9	Help users recognize, diagnose, and recover from errors <ul style="list-style-type: none">• Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution
10	Help and documentation <ul style="list-style-type: none">• Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.



Computer Science 312

Capstone Project

Assignment 5

Team Project		TOTAL MARKS / 50					
Team Number							
STUDENT NO.	STUDENT NAME	What contributed?					
		Possible Mark	Mark achieved				
			Excellent	Good	OK	Poor	0
Revise HCI portions based on feedback. Highlight changes.		10	10	8	5	3	0
Revise DB portions based on feedback. Highlight changes.		10	10	8	5	3	0
Revise SE portions based on feedback. Highlight changes.		10	10	8	5	3	0
Presentation		10	10	8	5	3	0
Demo		10	10	8	5	3	0
TOTAL		50					



Group 6

Members:

- Moegamat Tashreeq Waggie – Group Leader
- Jamie Lee Van Der Berg – Maps
- Khuliso Sikhwivhilu – Documentation
- Dahraan Abrahams - Designer
- Areeb Royeppen – Databases/Backend
- Dean Ockhuizen – Frontend/Backend
- Moegamat Ismaeel Ed'rees Jefferies - Documentation



Strut

What is it?

Presentation Overview

- 1 Human Computer Interaction
- 2 Databases
- 3 Software Engineering
- 4 Demo



Agile Process Model

- Scrum methodology
- Meetings
- Sprints
- Repetition

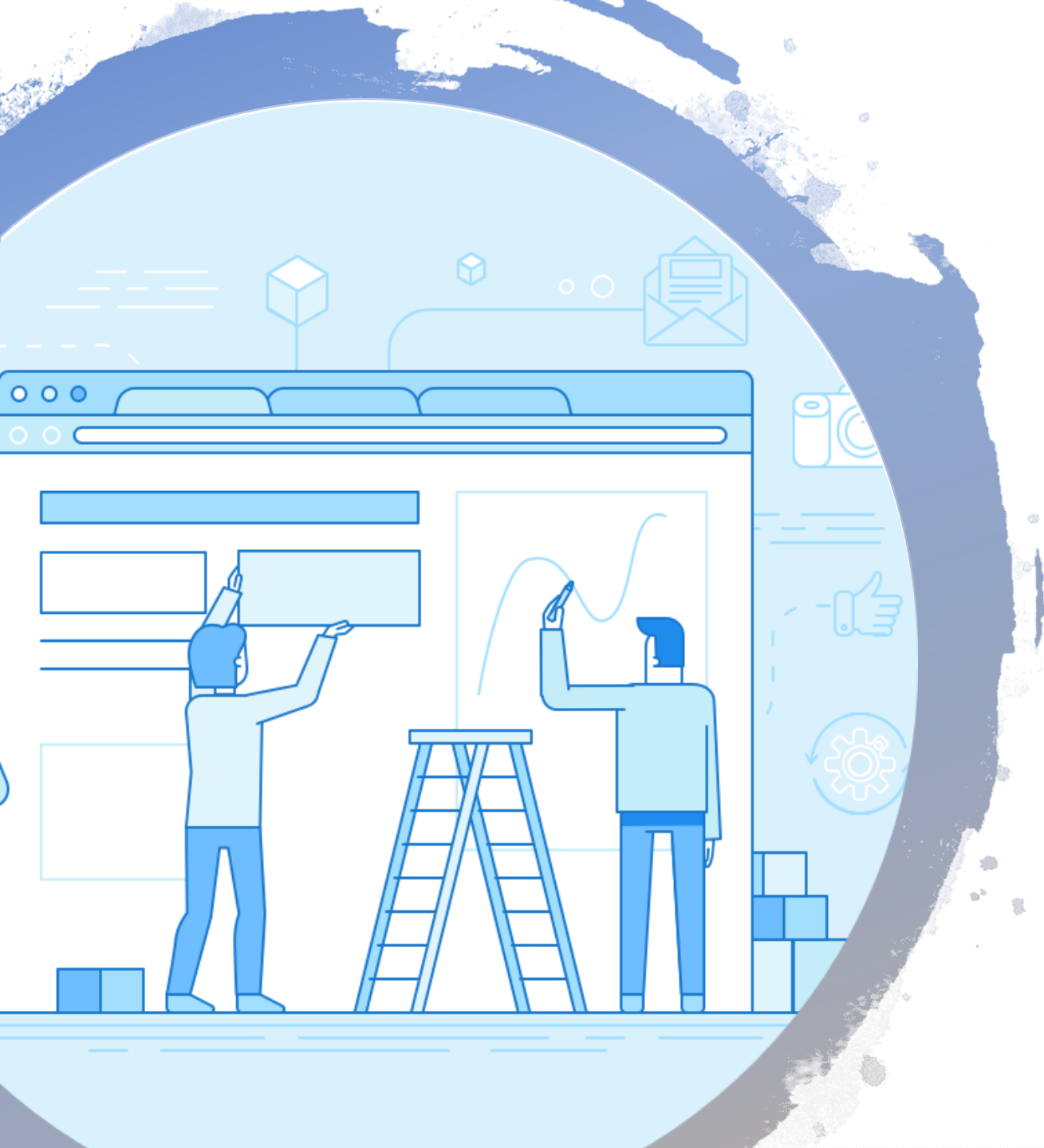


Fig. Agile Model

Figure 0 : Flow of the Agile Model

Human Computer Interaction





Prototyping

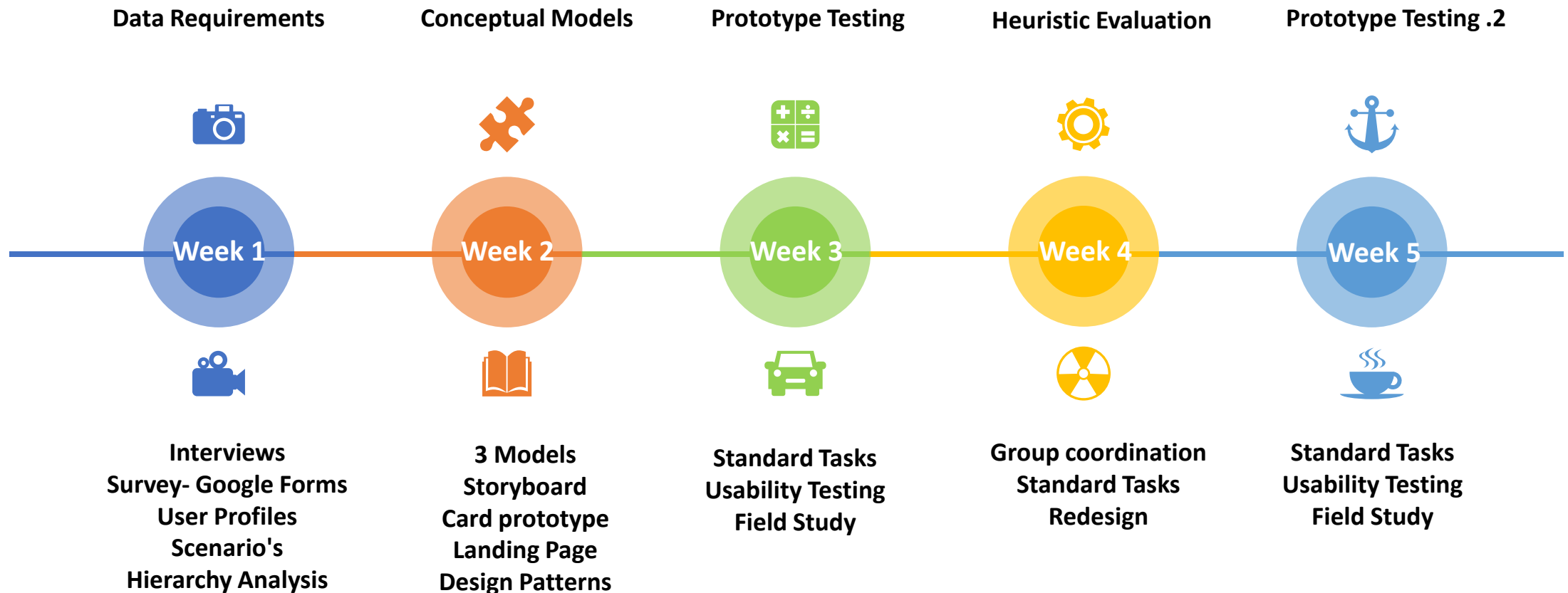
- Conceptual Models
- Storyboard
- Cardboard
- Product Prototype



Testing

- Usability Testing
- Heuristic Analysis
- Field Study

Human Computer Interaction Timeline



Databases





Django ORM
creates & manages
database models &
queries



Database Structure

- Student
- Enrolment
- Module
- Timetable
- Venue
- Building
- ExamTimetable
- BookedVenue

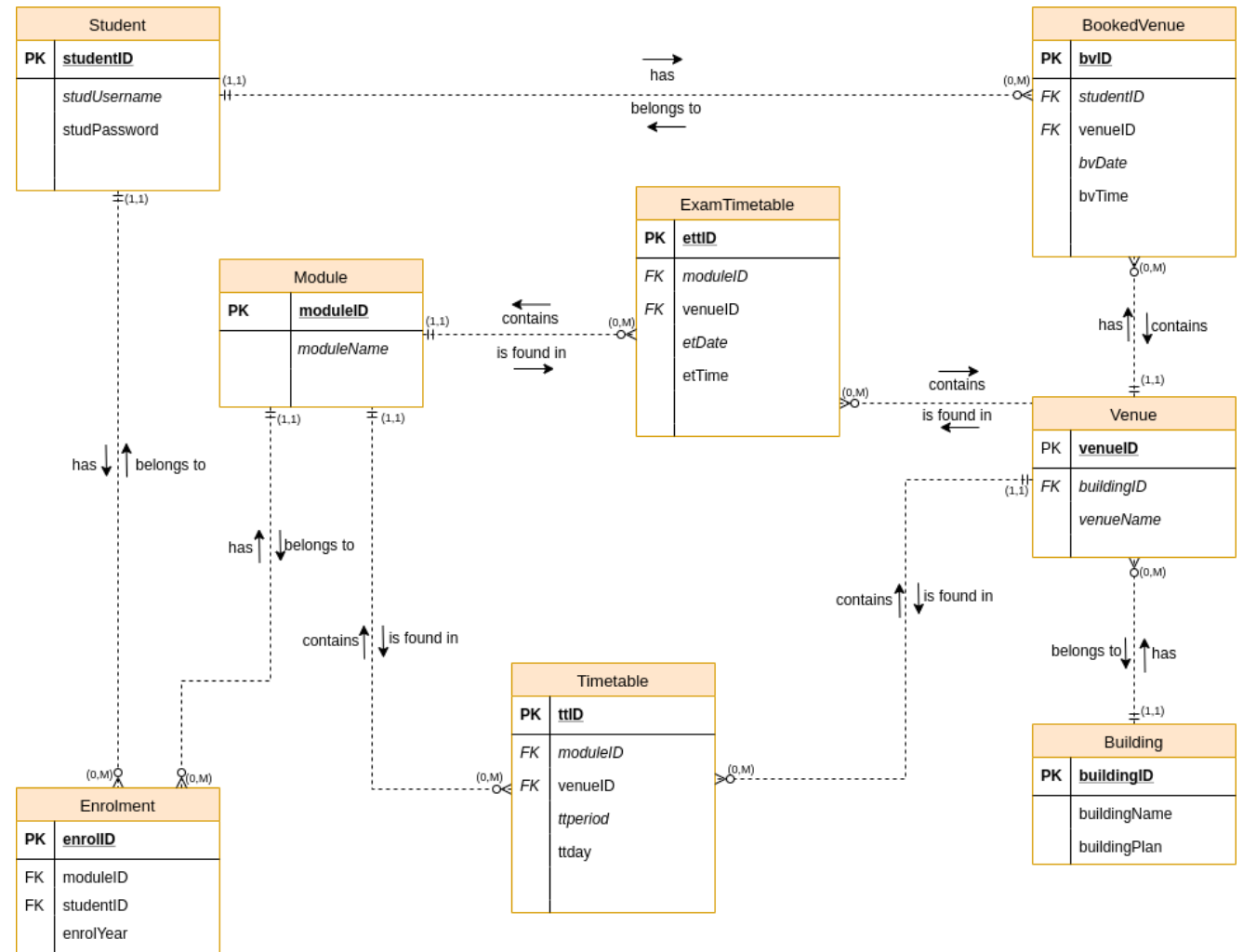


Figure 1 : This figure represents the structure of the Strut database.

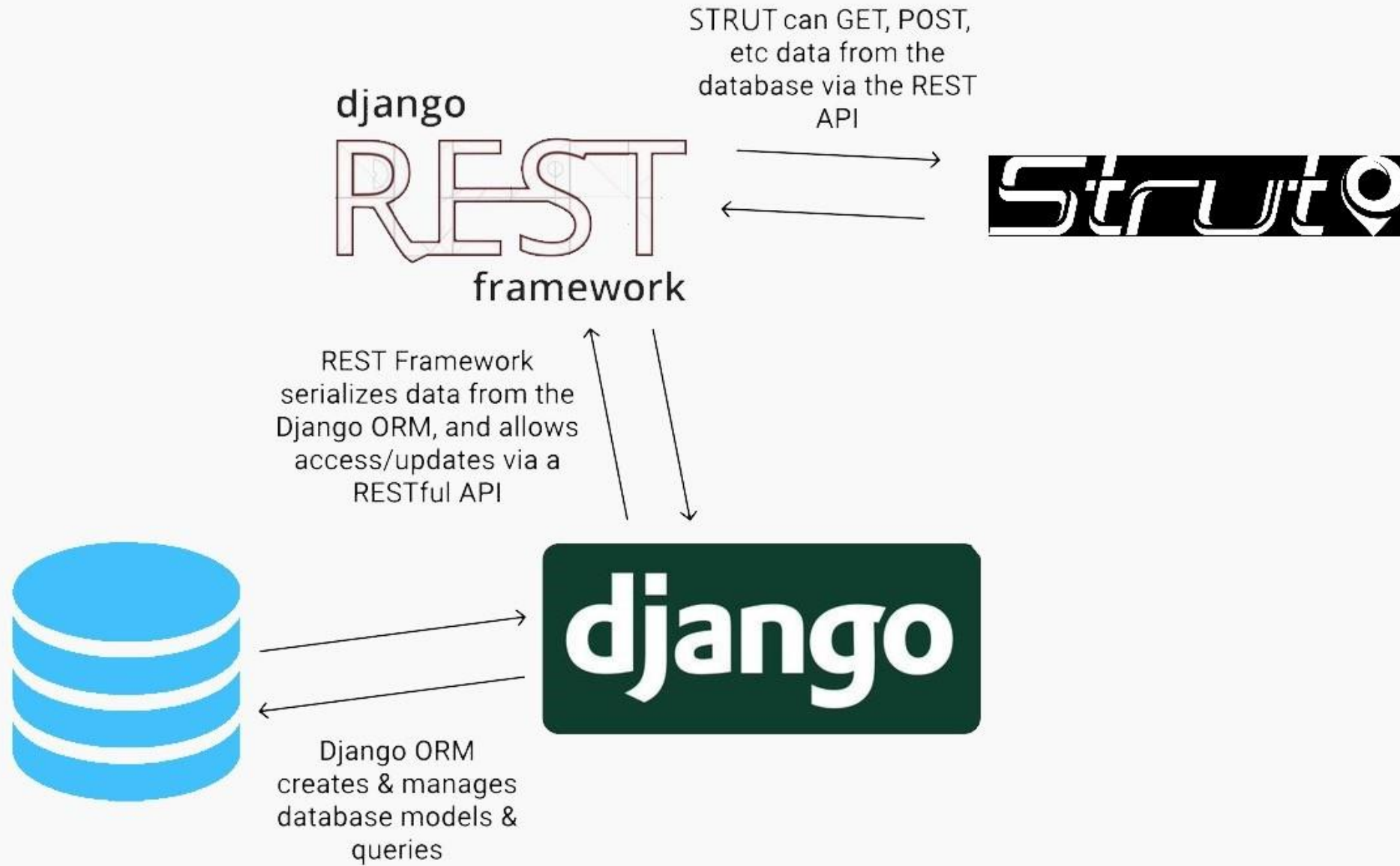
django
REST
framework

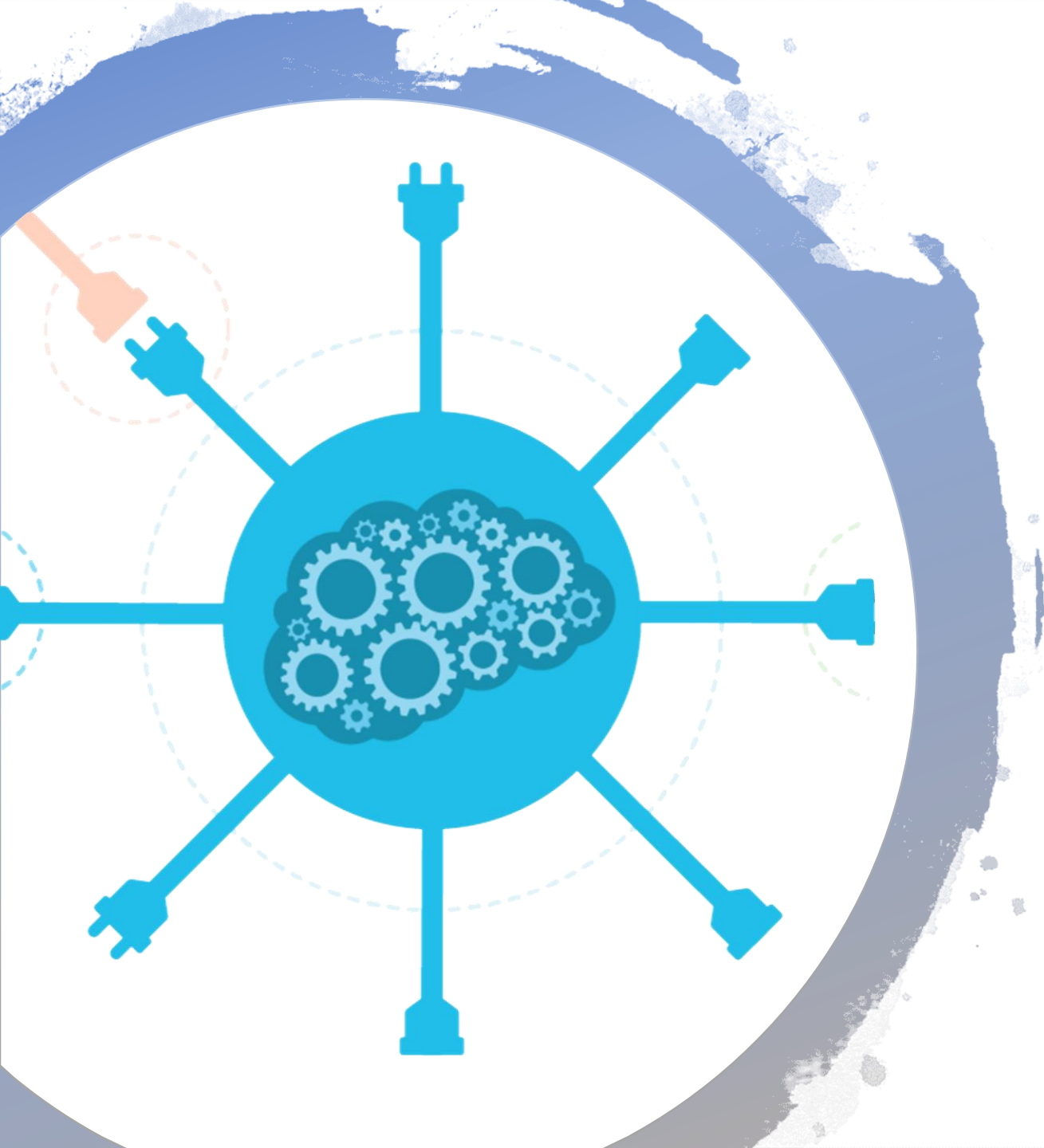
REST Framework
serializes data from the
Django ORM, and allows
access/updates via a
RESTful API



Django ORM
creates & manages
database models &
queries

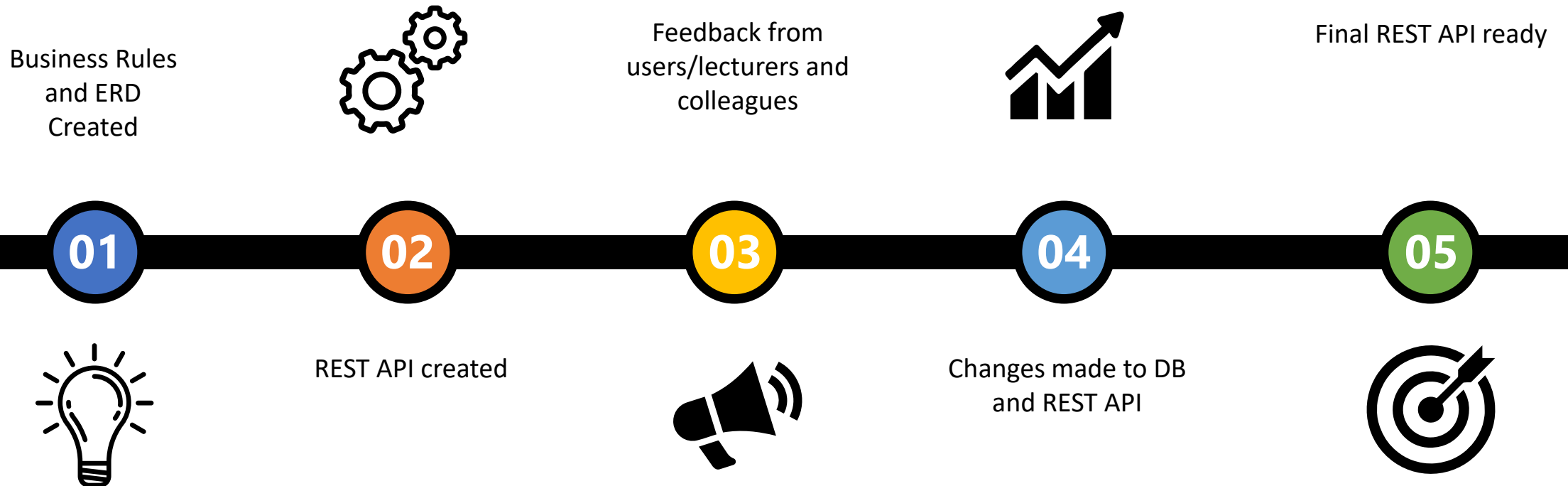







- /login
- /studentDetails
- /viewTimetable
- /viewExamTimetable
- /navigate
- /checkVenue
- /bookVenue

Databases Timeline



Software Engineering





Goals ,Principles and a Umbrella when it rains

- Principles
- Goals
- Umbrella Activities

Conclusion

- Constraints
- Future Iterations

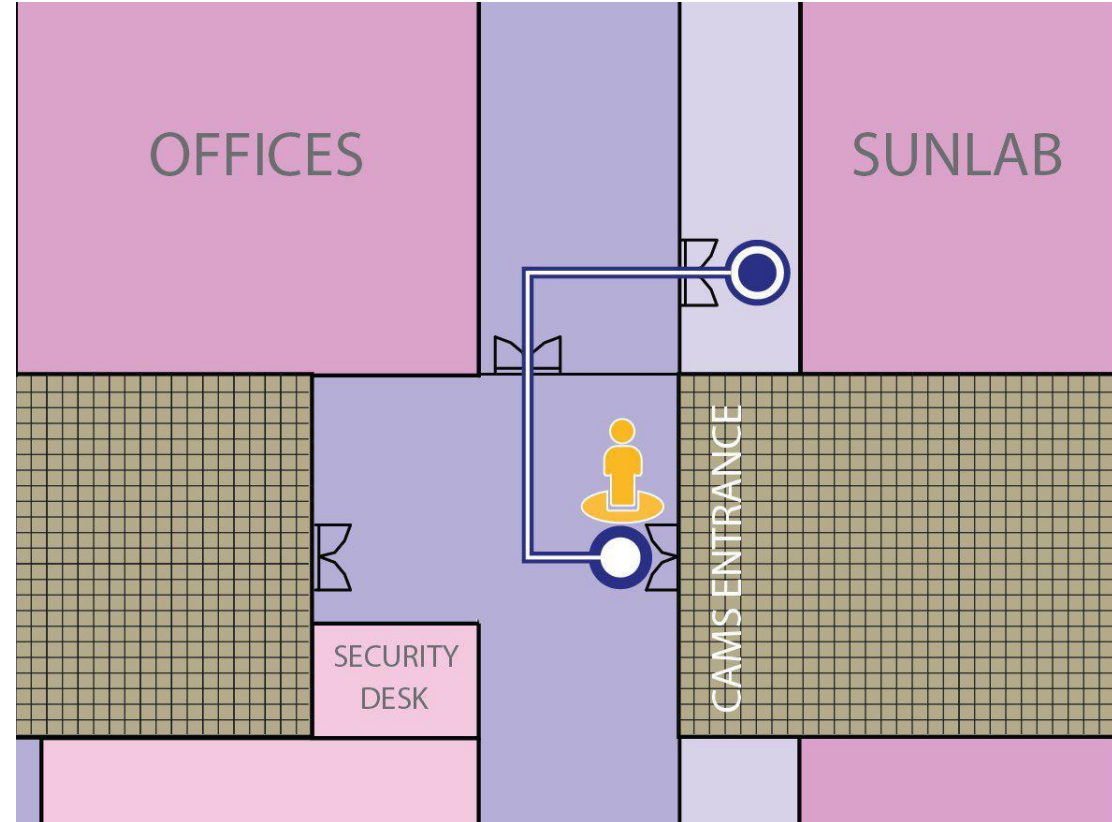


Figure 2 : In building navigation for strut

Demo

https://drive.google.com/open?id=1UUMI-dMO1p_0jmboxZjG0b-XI-4JQ5jIa

Video Presentation

<https://drive.google.com/open?id=1UN5MypFuNp6ObxN6JiWy9ootJtRNuE9u>