Capstone Project - Strut

Group 6

Assignment 3

Members:

Moegamat Tashreeq Waggie (group Leader) - 3437697

Jamie Lee Van Der Berg - 3762026

Khuliso Sikhwivhilu - 3744822

Dahraan Abrahams - 3772793

Areeb Royeppen - 3750662

Dean Ockhuizen - 3647526

Moegamat Ismaeel Ed'rees Jefferies – 3654954

Contents

Abstract

The following documentation is based on group 6 mobile application Strut. Strut is a mobile application that will less the burden of students. It will allow students to navigate campus to their desired venues without getting lost or needing to ask for help

Human Computer Interaction 1

Heuristic Evaluation

Consent form

Link to the consent forms - https://drive.google.com/file/d/1mGnCbV93ZkSgGa2E5n55TGyc3PuKrRU0/view?usp=sharing

Link to the signed consent forms - https://drive.google.com/drive/folders/16eLDOooHzItt-THEcPbJlAbGSCwf_bt8?usp=sharing

Briefing

Link to audio - https://drive.google.com/file/d/123ojXAe93U3bNnMej4Lbt2QdivUuRw-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.

Link to the written briefing - https://drive.google.com/file/d/1hbCr5tdX2G-h4RXw6No3TxH-8skLCVJ5/view?usp=sharing

Evaluation Period

Complied 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 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

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. The back-button has a familiar design, making the recognition of its function quick.
When selecting the 'VISITOR' profile-type, the user will be directed to the 'DIRECTIONS' page. This page has a simple design with only 3 options on the page. This makes recall of the functions of the options of this page easy and ultimately the campus-navigation for visitors easy. When selecting the 'FIND' option the users are then taken to the page displaying the path they are to take between their entered points. This page has a very simple design and a similar design to the most common navigation app - Google Maps-, only displaying the path on the campus map, a back-button, and the details of the path in words. This makes the recognition of the information displayed on the page easy.
When selecting the 'STUDENT' profile-type, the user will be directed to the login page. This page has a simple design and familiar design, the design has high degrees of the elements of practice, recency and context incorporated into its design. The recall of the login-page functions via the buttons is then easy. Upon login, the student-user will be taken to a homepage with many options and a back-button. All of these options, when selected, either display simple and straight-forward information, or directs the user to the 'DIRECTIONS' page mentioned previously, or to the 'BOOK VENUE' page which maintains the same

design as the 'DIRECTIONS' page. Hence the overall design from the homepage is simple and familiar, making recall of the functionality of the options easy.

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 bull 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

Human Computer Interaction 2

Redesign and document 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

Problems addressed in the redesign of the prototype

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.

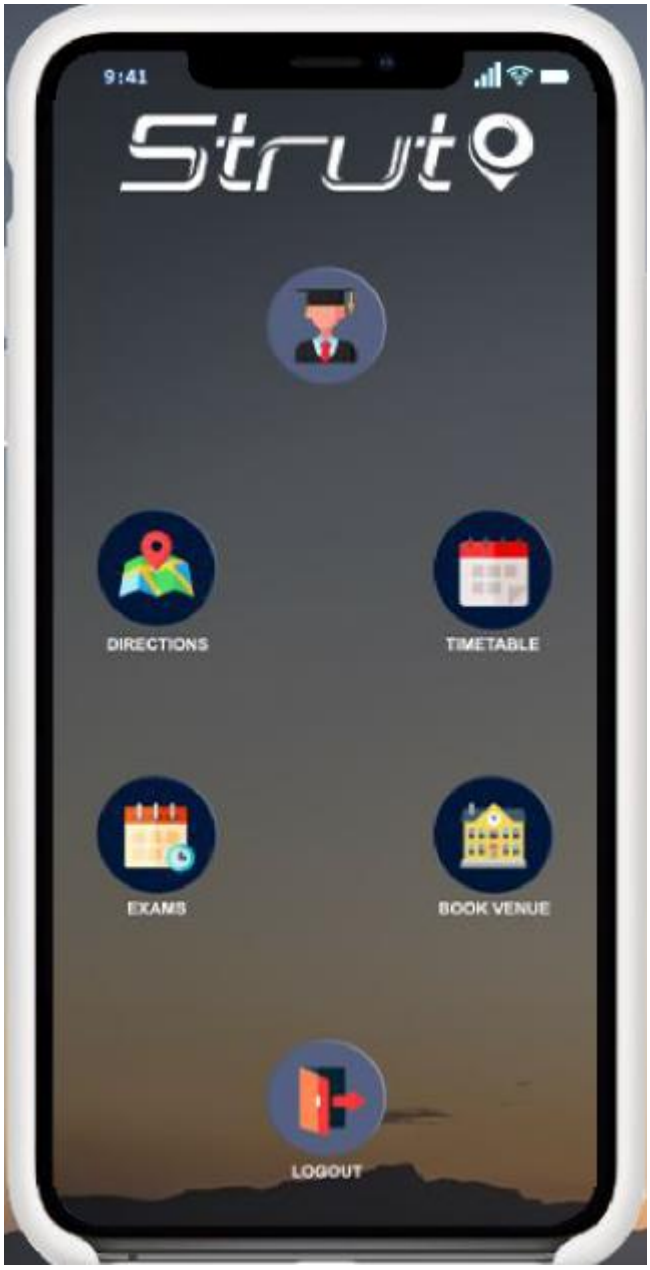
3. The logo has become a home button.

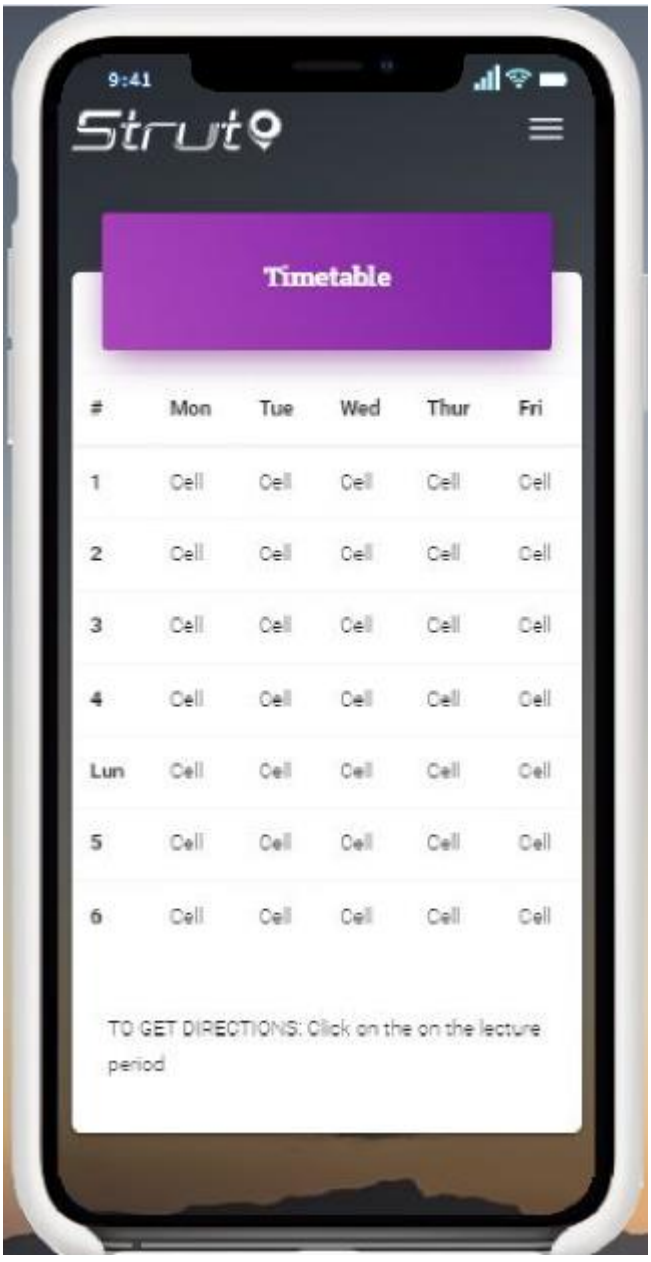
   This was done to accommodate the drop-down menu and that if the user our like to go back to the home screen they can do it using the logo.

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





Human Computer Interaction 3

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.

Human Computer Interaction 4

Comparison Table

	Usability Testing	Field Studies	Heuristic Evaluation
Findings	• The prototype used in testing allows developers to recognize problems that the user experiences in a controlled setting • We did this controlled setting in the Sun Lab last week. • We were able to get real task problems from the experiment.	• Users were using their phones while on the move more than being on a computer in a room. The Strut prototype was accessible from there smartphones by just being provided a link • Friends, classmates and family member were able to do this in their own time. They had more freedom than usability testers	• Having experts (such as our fellow HCI classmates) use their experience of user knowledge and following heuristic methods makes for a careful approach to development as they go through tasks. • Experts are able to find low cost problems and potential problem users will have.
Benefits	• Users are in a controlled setting. • Users can be timed and recorded. • Hypothesis could be tested. • Data could be analysed • Environmental influences are being controlled.	• Observe how product performs in the 'real world' • Setup can be quick and easy • Cost efficient • No suitable environment needed	• Few ethical and practical issues. • It is a detailed process that assesses the product against a clear criteria • Tends to focus on more relevant areas it identifies to be important • Average of 5 evaluators allow identifying 75%-80% of usability problems
Costs	• Finding a suitable room/environment such as the sun lab. • Find suitable candidates/users such as our friends that study with us. • Making time for tests.	• There are environmental and social influence • No control over users.	• Important problems can be missed • Many trivial problems are often identified • Experts could be biased • Problems often identified might not/are not critical.
Limitations	• Time is not always suitable, • Students/Candidates are not always available, they might not be at campus or has class to attend	• Users cannot be recorded • Only can collect limited data	• Experts may be difficult to find. In our case we were only able to get a hold of 4 members of group 2 because of time • The experts might/are costly.

Software Engineering

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.

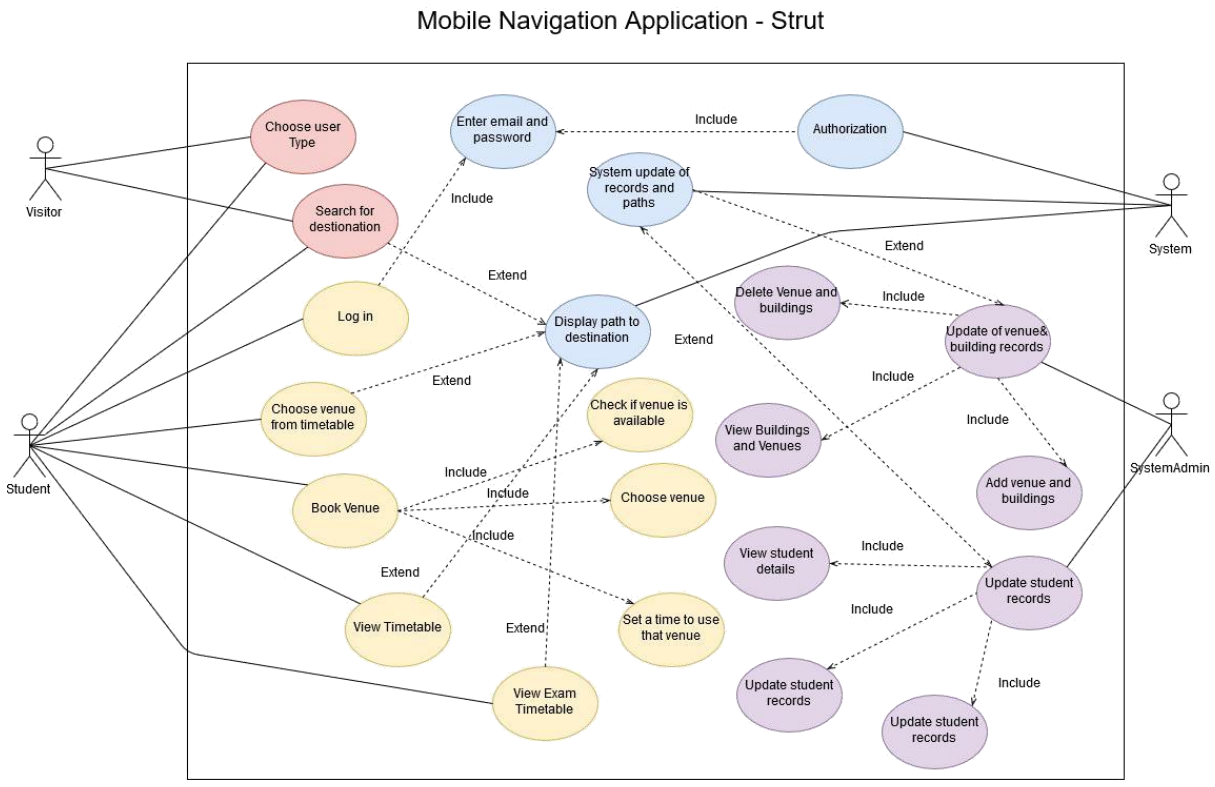
- Software Configuration Management

  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.

UML Diagram



Databases

Internal model in MySQL

```
CREATE DATABASE strutDB;

CREATE TABLE Student (
        studentID int NOT NULL AUTO_INCREMENT,
        studUsername varchar(255),
        studPassword varchar(255),
        PRIMARY KEY (studentID)
);
CREATE TABLE Building (
        buildingID int NOT NULL AUTO_INCREMENT,
        buildingName varchar(255),
        buildingPlan
        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 NUL AUTO_INCREMENTL,
        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),
	FOREIGN KEY(venueID),
	CONSTRAINT UC_Timetable UNIQUE (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),
	FOREIGN KEY(venueID),
	CONSTRAINT UC_ExamTimetable UNIQUE (venueID, etperiod, etDate)
);
CREATE TABLE BookedVenue (
	bvID int NOT NULL AUTO_INCREMENT,
	studentID int,
	venueID int,
	bvDate DATE,
	bvTime int,
	PRIMARY KEY(bvID),
	FOREIGN KEY(studentID),
	FOREIGN KEY(venueID)
);
```

Friday 18 October 2019

Computer Science 312
Capstone Project

Assignment 4

Team Project **Team Number**			**TOTAL MARKS** **/ 40**
STUDENT NO.	**STUDENT NAME**		**What** contributed?

	Possible Mark	**Mark achieved**				
		Excellent	*Good*	*OK*	*Poor*	*0*
HCI1. Heuristics and evaluation of one prototype.	**4**	4	3	2	1	0
HCI2. Redesign and document what has changed and why.	**4**	4	3	2	1	0
HCI3. Compare the findings from heuristic evaluation with those from the previous assignment (Assignment3). Differences and preferences.	**4**	4	3	2	1	0
HCI4. Comparison Table	**8**	8	8	4	2	0
DB1. Internal model in MySQL, data types for attributes.	**10**	10	8	5	3	0
SE1. Document the application of SE principles, what has changed, . . . etc.	**10**	10	8	5	3	0
TOTAL	**40**					