EGR 401 – Capstone Design

Deliverable 5: Design Architecture, Detailed Design, and Detailed Budget

Team Name: Blueprints

Team Members: Jordan Ziegler

  Jeremy Fischer

  Justus Karenzi

  Sudi Nsengiyumva

  Dylan Shanahan

Client’s Name: Dr. Mark Gordon

Advisor’s Name: Professor Thomas Renck

Version: 1

Date: November 10, 2014

**5.1 Design Architecture**

**Level 0 Design**

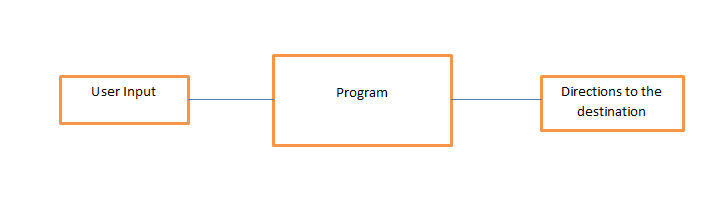


Figure 5.1-Level 0 software architecture design

|  |  |
| --- | --- |
| *Module* | Program |
| *Input* | User Input |
| *Output* | Directions to the destination |
| *Functionality* | The augmented reality app will help students locate the rooms/offices. The user will input the room through a drop-down menu and the app will direct the user to his/her destination |

Table 5.1-Level 0 Input/Output software Architecture

Our level 0 design includes all the modules that we will need for the project. The user will be able to choose a room from a drop down menu. Upon retrieving the room from the database, the app will help the user navigate to the corresponding room. Below in table 5.2 are all the modules with their respective inputs, outputs and functionalities.

|  |  |  |  |
| --- | --- | --- | --- |
| *Module* | *Inputs* | *outputs* | *Functionality* |
| **UI** | CSV | Drop down menu | The UI will provide the user with a friendly experience |
| **Drop Down Menu** | User Input | Directions | Easily select room |
| **Map Selection** | user input from drop down menu | map of building | This module will interact with the map of the building and retrieve the directions |
| **Image Processing** | Picture taken from device camera | Recognition of user location | Recognize the picture from camera to determine location of device |
| **Directions** | Current Location | User destination | Helps user get to their destination |

Table 5.2 Design Architecture Modules

**Level 1 Design**

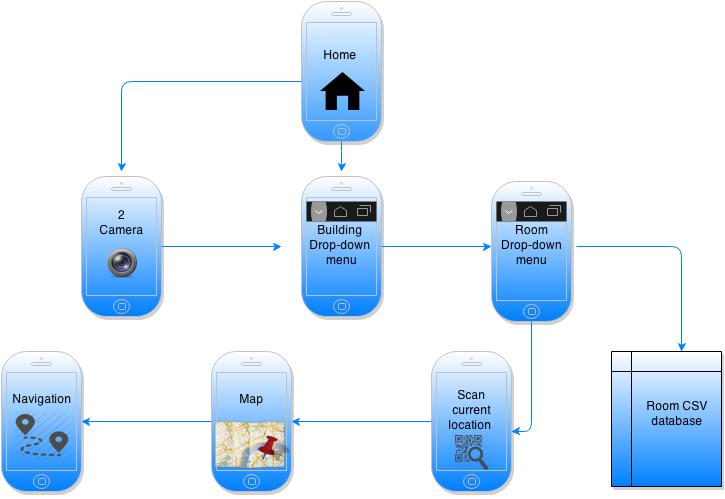
****

Figure 5.2-Level 1 Interface and Functionality

5.2.1: User Interface (UI) Module

The above interface and functionality flowchart illustrates the way the user will communicate with the application. Basically, the user will open the app and get to the home page which will consist of a camera and building drop down. After the user select the building that he/she wants to get to, the room dropdown menu appear so that one would choose which room to go to. When the user finishes choosing the room the app will look into csv file from the database and pull up the location of the room on the map. At that point the user will need to scan his/her current location so that the navigation would start to direct him/her. The user will have an option of closing the app once the destination has been reached.

**5.2 Detailed Design (Drawings) and Calculations**

**General Software Flow Chart and Explanation**

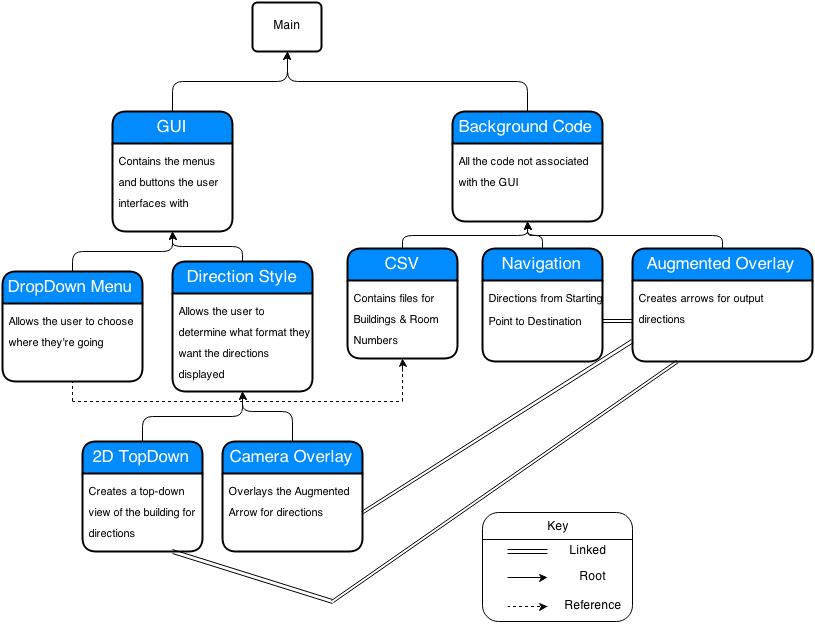


Figure 5.3 - General Software Flow Chart

The general flow diagram, shown above in Figure 5.3 shows the main parts of the program. The main program is divided into two sections, the GUI and background code. The GUI is the user interface and is subdivided into the different modules that the user interacts with. The user determines what building and room they are going to using the drop down menu. They also have the choice of output format, being able to choose between the augmented reality format and the more traditional 2D top down map picture. The background code is essentially all the code that happens behind the scenes, which the user does not interact with. This includes things like the CSV file, the navigation software, and the augmented overlay. The CSV file is where the words in the drop down menu come from. The navigation software controls the algorithm which decides what directions the program gives the user. The Augmented overlay is the images that are placed on the screen in order to provide the user with the navigation.

|  |  |
| --- | --- |
| *Module Name* | Main |
| *Module Type* | Coordination |
| *Input Arguments* | None |
| *Output Arguments* | None |
| *Descriptions* | The main function is the entry point for the program. It starts the GUI and awaits user input. Once the input is entered it corresponds with the Background Code to implement calculations and additional work functions. |
| *Modules Invoked* | GUI, Background Code |

Table 5.3: Main

Table 5.3 shows the specifics for the Main module. The Main module controls the flow of the program. It shares information between the GUI and Background Code to make the program work. This module is designed for coordination between separate functions and to transfer information between those functions.

|  |  |
| --- | --- |
| *Module Name* | GUI |
| *Module Type* | Input and Output |
| *Input Arguments* | Design code |
| *Output Arguments* | Attractive menus and easy to use features |
| *Descriptions* | This will make the app easy to use and attractive to the user |
| *Modules Invoked* | DropDown Menu, Direction Style |

Table 5.4: GUI

Table 5.4 shows some of the specifics of the graphic user interface(GUI) module. The user interface is primarily input and output, calling the drop down menu and direction style. This module is primarily designed for attractiveness and ease of use.

|  |  |
| --- | --- |
| *Module Name* | DropDown Menu |
| *Module Type* | Input and Output |
| *Input Arguments* | User selected building/room |
| *Output Arguments* | Destination |
| *Descriptions* | Easy way for user to select their location |
| *Modules Invoked* | None |

Table 5.5: DropDown Menu

Table 5.5 shows some of the specifics of the drop down menu. The menu is primarily input and output. This module is primarily designed for ease of use, and also to allow the user to select where they are going.

|  |  |
| --- | --- |
| *Module Name* | Direction Style |
| *Module Type* | Input and Output |
| *Input Arguments* | User desired style |
| *Output Arguments* | Either the 2D TopDown styler, or the 3D Camera Overlay |
| *Descriptions* | Provides the user with two different options of how they can reach their destination |
| *Modules Invoked* | 2D TopDown, Camera Overlay |

Table 5.6: Direction Style

Table 5.6 shows some of the specifics of the direction style module. This module is primarily input and output. This module allows the user to determine whether they want a 2d or 3d display. This module invokes both of those options depending on what the user picks.

|  |  |
| --- | --- |
| *Module Name* | 2D TopDown |
| *Module Type* | Input and Output |
| *Input Arguments* | Directions |
| *Output Arguments* | 2D top-down view of building floor overlayed with directional arrow |
| *Descriptions* | Module to help user see the layout of the floor and to easily interpret directions |
| *Modules Invoked* | None |

Table 5.7: 2D TopDown

Table 5.7 shows some of the specifics of the 2D TopDown module. This module allows the user to see the map of the building from above. This module shows the directions much like Google Maps does.

|  |  |
| --- | --- |
| *Module Name* | Camera Overlay |
| *Module Type* | Input and Output |
| *Input Arguments* | Image captured from environment |
| *Output Arguments* | Display on the device’s screen |
| *Descriptions* | Captures images and displays them onto the screen of the device |
| *Modules Invoked* | None |

Table 5.8: Camera Overlay

Table 5.8 shows some of the specifics of the Camera Overlay. This module allows the user to see the image captured from the environment. This image is displayed on the device’s screen for the user.

|  |  |
| --- | --- |
| *Module Name* | Background Code |
| *Module Type* | Input and Output |
| *Input Arguments* | Images from camera, CSV file information |
| *Output Arguments* | Calculations and Directions |
| *Descriptions* | This code calculates where the user needs to go, and which direction they should take based on the captured image and input destination |
| *Modules Invoked* | CSV, Navigation, Augmented Overlay |

Table 5.9: Background Code

Table 5.9 shows some of the specifics of the Background Code. The module takes images from the camera, and the CSV information in order to develop the calculations and directions the rest of the code requires in order to develop the output of the program. This code calculates how the user gets to where they need to go. This module invokes the CSV, Navigation, and Augmented Overlay modules.

|  |  |
| --- | --- |
| *Module Name* | CSV |
| *Module Type* | Input and Output |
| *Input Arguments* | None |
| *Output Arguments* | Building/floor , Room number, and room description |
| *Descriptions* | Method of formatting the data used to find each room in a building |
| *Modules Invoked* | None |

Table 5.10: CSV

Table 5.10 shows some of the specifics of the CSV module. This module is given by the programmer/administrator and outputs the building, floor, room number, and room description. This module includes the information necessary for the drop down menu.

|  |  |
| --- | --- |
| *Module Name* | Navigation |
| *Module Type* | Input and Output |
| *Input Arguments* | Current user location |
| *Output Arguments* | Path the user should take to get to destination |
| *Descriptions* | The process of determining user position, and routing them to their destination |
| *Modules Invoked* | None |

Table 5.11: Navigation

Table 5.11 shows some of the specifics of the Navigation module. This module is given the user’s current location and takes that to develop a path that the user should take to get to their destination.

|  |  |
| --- | --- |
| *Module Name* | Augmented Overlay |
| *Module Type* | Input and Output |
| *Input Arguments* | Images from camera |
| *Output Arguments* | Overlay arrow displayed onto screen based on the users destination |
| *Descriptions* | Displays an arrow that will navigate the user to their destination |
| *Modules Invoked* | None |

Table 5.12: Augmented Overlay

The module shown in Table 5.12 is the Augmented Overlay module. The image from the camera is taken and the augmented reality (directions in the form of arrows) is overlaid on the image. This module takes the image and uses to determine where to put arrows which tell the user where they’re going.

**5.3 Updated SOW and Project Management Plan**

Our statement of work goes through what the team is currently working on, and the current standpoint on the project’s completion.

1. Currently working on Deliverable 5
   1. Design Architecture
   2. Detailed Design
2. Performing background research to solidify our understanding of the SDK and Android programming.
3. Researching relevant technology to find out what types of features are applicable to the app, and choose which types of features we plan on implementing.
4. Learning the Javascript programming language that is associated with the Vuforia SDK.

**5.4 Detailed Project Budget**

Total Purchasing Budget: $500

Capital Equipment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item Description** | **Item No.** | **Company** | **URL** | **Unit price** | **Quantity** | **Subtotal** |
| Google Nexus 7 Tablet | N/A | Amazon | <http://www.amazon.com/Nexus-Google-7-Inch-Black-Tablet/dp/B00DVFLJDS/ref=sr_1_1?s=pc&ie=UTF8&qid=1415662359&sr=1-1&keywords=google+nexus+7> | 179.00 | 1 | 179.00 |
| Vuforia SDK | N/A | Qualcomm | <https://www.qualcomm.com/products/vuforia> | FREE | 1 | FREE |
| Total | 179.00 |  |  |  |  |  |
| Budget | 500 |  |  |  |  |  |
| Surplus | 321.00 |  |  |  |  |  |

Table 5.13: Capital Equipment - Bill of Materials (BOM)

Administrative Supplies: None

Reference Materials: None

Staffing

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Hours** | **Salary ($ per hour)** | **Subtotal** |
| Jeremy Fischer | 60 | 20 | 1200 |
| Dylan Shanahan | 71 | 20 | 1420 |
| Jordan Zielger | 64 | 20 | 1280 |
| Justus Karenzi | 61.5 | 20 | 1230 |
| Sudi Nsengiyumva | 60 | 20 | 1200 |
| Thomas Renck (Advisor) | 3 | 40 | 120 |
| Total | 6450 |  |  |

Table 5.14: Payroll