# **Design choices and structure**

User friendly navigation system for lecturers and administrators to store or validate user input. Specific colours and labels to describe each part of the system and user requirements for functionality that will. User identification is required to acquire information that will be stored or verified. Functions that allow user input for lecturers’ claim and a submit button to allow claims to be stored. Stored inputs should be accessible to administrators for validation and restored with edited information which consists of verification status. Each window or function should be linking to another window through the navigation system, e.g., using buttons as a link to navigate to the next.

The Contract Monthly Claim Database should contain all functional requirements of the system, which will be used to store data. The data should have gone though necessary grouping using other functions and forms to describe the group it belongs to. All Entities of the Database should relate to each other as data from each entity will be used in another. The data will be stored in attributes that are grouped by entities.

The GUI should consist of buttons that contain clear and easily readable labels that allows users to easily understand. Each window should contain visual hierarchy with constraints that prevent error inputs or errors in the system. The structure of each window should be consistent and meet design standards. A simple, user-friendly interaction design that caters to different issues. Micro interactions that show functionality with a lack of errors. Popular font styles that users are accustomed to that are unambiguous. The GUI should be understanding customer needs and expectations, prioritising customer satisfaction and improving.

Assumptions:

* Meeting deadlines
* Redundant data
* Might contain potential errors
* Ambiguous requirements of user input

Constraints

* Scheduled tasks
* Functions with conditions
* Tests
* User-friendly

# **Project Plan**

Planning project [5 days]

* Detailed explanation and requirements of project that can be used to reference for steps of design
* Gaining objective and understanding purposes of each activity of the project
* Proper scheduling for each task for easy work

Build prototype with design features (the prototype must not be functional) [2 days]

* Creating prototype for system that can be used as a starting point to develop system

Add functions [2 days]

* Function to store claim from lecturers
* Function to link pages in a navigation form
* Function that reduces redundant data
* Function that makes data reliable
* Function that stops any potential errors

Edit prototype [2 days]

* Fix any errors
* Add new features for easier user interactions
* Add any required design forms
* Reduce unambiguous designs and functions

Testing [1 day]

* Take program through unit tests
* Fix any errors
* Make sure program works on deferent devices