**Life Cycle Architecture Milestones**

Assignment 2

Group 2:

* Sobana Handi Achini Thisarangi De Silva
* Dehemi Vihara Dissanayake Liyanage
* Gaury Chethana Thanthirigama
* Temuulen Tsengel
* Michael Weisang

Deliverables:

1. Project Vision
2. Domain Class Model
3. ERD Diagram
4. Use Case Description and model
5. Non-Functional Requirement Specification
6. Architectural Notebook
7. Risk List
8. Master Test Plan
9. Homepage
10. Registration page
11. Login page
12. UAT
13. Project Plan
14. Inception Phase Status Assessment

Remote Mental Health Management

“Naaman” Vision

# Introduction

During the Covid era of lockdown, there is a lot of concern emerged. One of the main concerns is social interaction. Human is homo socius, which means Human is a social being. Humans need to interact with the other and the lockdown makes it difficult of making their needs. Even with the internet and social media existing, it is not enough for the majority of people. This has raised a lot of issues in the mental health state of the people.

Mental health is a really important factor for people to function. Mental health affects our way of thinking, feeling, and action. Mental health is important for people to cope with stress, interact, and making decisions. Having low mental health can lead to a lot of detrimental things. For example, mood swings, low energy, no motivation, food binge, self-harm, etc. All this strait is disadvantageous to society and can lead to suicide.

To tackle this issue, Melbourne Hospital has decided to develop a web-based social media platform and assign the task to Group 10 "Runtime Terror". The project will be called Project "Naaman". Naaman name came from a story from the bible. Naaman is an army commander who developed leprosy disease. He was sent to a great prophet Elisha and expected to be blessed and cured of this disease. But the prophet has just sent a messenger and told Naaman to wash in the Jordan river instead of a blessing. He was angry and dejected that he didn't get the miraculous blessing or some sort of ritual. In the end, he was persuaded by his companion to wash as it was a simple action. Naaman is cured after washing. The moral of this story was that instead of waiting for some miracle to happen, sometimes a simple action advised by the expert is all you need.

Project "Naaman" aims to help decrease the mental health issue by providing a social media platform where people can share their stories and get online consultation from an expert. It provides an accessible and safe environment where people can share their concern. Due to the nature of mental health issues being a private and sensitive concern, Naaman will provide an anonymous platform so that people can freely express their concerns without being judged by close relatives or friends if using their usual social media.

# Positioning

## Problem Statement

## Table Description automatically generated

People that are social distancing is facing an issue of declining mental health due to lack of social interaction. This has led to low motivation and mood which cause low productivity and some suicide tendency. To solve this problem, there is a need for people to vent their frustration or complain they have without facing some judgement or backlash, and also receive a professional care to help with the issues.

## Product Position Statement

Table

Description automatically generated

Naaman is created to provide a social media platform to help with people who have low mental health. The platform will provide a place for people to vent and share theirs stories and also get professional help. Naaman is different than other social media because it have an anonymous feature so people can share their stories or complain without getting backlash from people involved or some friends or relative.

# Stakeholder Descriptions

## Stakeholder Summary

Table

Description automatically generated

## User Environment

Text

Description automatically generated

To protect the users, the user needs to create an account with password and authentication to prove their identity. The user can also decide the level of information that can be accessed by the others. The low setting being everyone can access it, and the highest setting where only the assigned physicians can see it.

There will be a different interface where the physicians can use it. To access the interface, the physicians need to create an account and given access by the admin. The interface will include an extra function to help the physicians managing their patient.

# Product Overview

## Needs and Features

Table

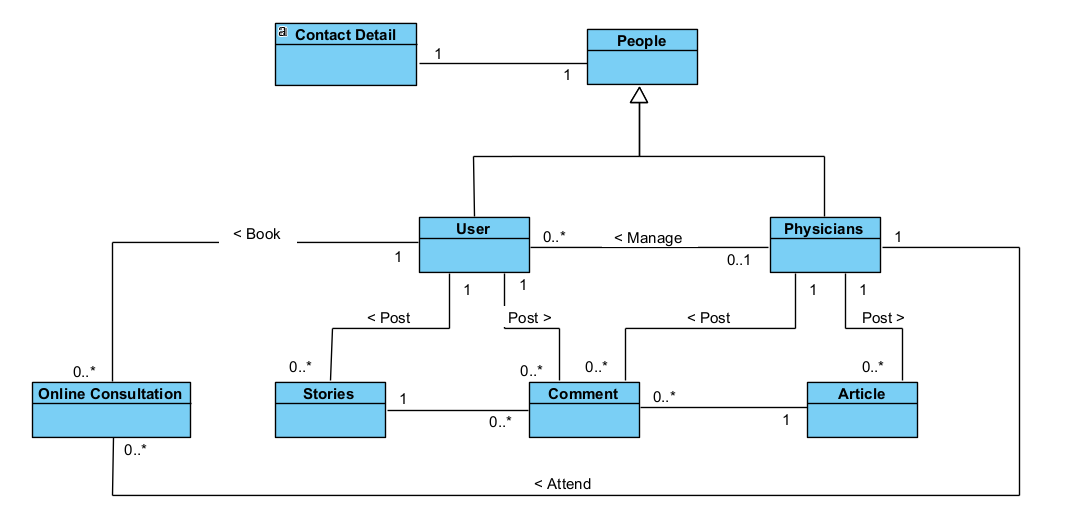
Description automatically generated

# Other Product Requirements

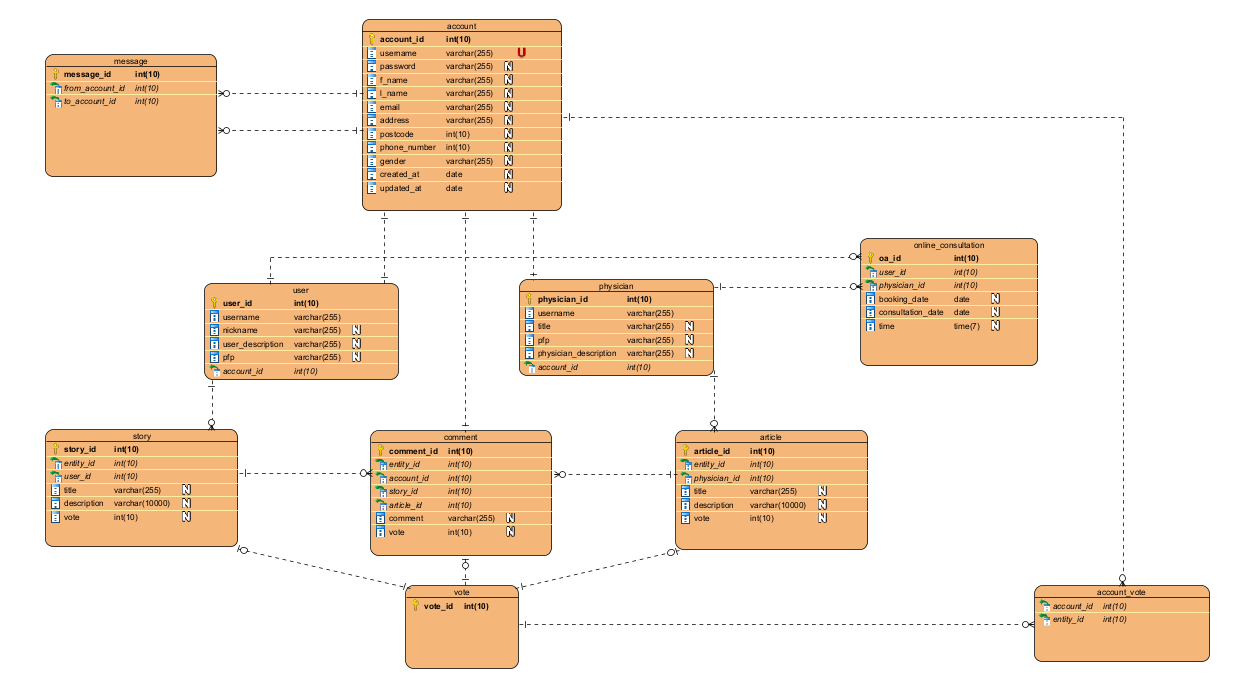
Table

Description automatically generated

Domain Class Model



ERD diagram



# Mental Health Management system (Naaman) public page

Diagram

Description automatically generated

Use case description

This use case shows the public page of remote mental health management system. Mainly the user can visit the mental health management system. Register or logging to the system and post articles and stories. There are main functions in the mental health management public page. (Public page must be news feed page because the new user is able to get an overall idea about the system) while moderator can log into the system and control user behaviours. the selected web project is an open public project. It will be publicly used by different people in every nook and corner. We are unable to predict their behaviour. That is the main reason for using a moderator for monitoring. Additionally, posts of new users will be hold automatically and they will be reviewed by the moderator and publish them. At the beginning of the project, this moderator is a human being. However, as a future suggestion the moderator should be an artificial intelligence system because of side traffic.

# Use-Case: <Access the web >

1 Brief Description

This use case allows to access the Remote Mental Health Management system public webpage with selected strong and valuable article and stores for the user. (Mental health management system public website must be with post sharing page)

2 Actors

2.1 User

2.2 System

3 Pre-Conditions

* 1. The user must be a new user or have not logged in to the system
  2. User used browser hasn’t Mental health management system cookies
  3. honeypots must be online and Website must be online (server-side honeypots are used as a security method which is used by administrators for prevent attacks, avoid hacking, identify malicious activities.

4 Normal Flow

Table

Description automatically generated

5 Alternate Flows

if the user registers to the system and user logout the system

Table

Description automatically generated

6 Sub flows

Request with wrong web address

Graphical user interface, text

Description automatically generated

7 Key Scenarios

New and registered users visit Mental health management system public web page first. The admin panel in this project decide considering about web site security therefore original web server IP address must be hide. Config the honeypot and user access the system withing the honeypot in this configuration can prevent the unauthorized access. While moderators can check the Mental health management system website security issue. The moderators can control the user behaviours and control the website

if user is new, system generates website content using his geological location data. if user is currently registered, using his past data for generating the website content.

8 post-conditions

Add new record in to the user log with login IP address

Create a cookie for the user

9 Special Requirements

Graphical user interface, text, application

Description automatically generated

# Use-Case: < Register New User>

1 Brief Description

This use case allows to register new user into the Mental Health Management system. In this stage system check the user’s browser cookies because if browser has any cookies related to the Mental health management system automatically login as a cookie’s related user.

2 Actors

2.1 User

2.2 System

3 Pre-Conditions

3.1 The user must be new user

* 1. Website must be online
  2. User’s browser hasn’t Mental health management system’s cookies

4 Normal Flow

The use case begins when user access web page (home page)

Table

Description automatically generated with low confidence

5 Alternate Flows

if new user visits the website and try to comment, make a post or other tasks.

Table

Description automatically generated

6 Sub flows

Table

Description automatically generated

7 Key Scenarios

Graphical user interface, text, application, email

Description automatically generated

# Use-Case: < User Login>

1 Brief Description

This use case allows the user to login into the Mental Health Management system. If user tries to login from user’s previous device and it has some cookies related to the Mental health management system, system automatically login to the system as cookies related user. (Cookie’s expiry dates also check in this stage)

2 Actors

2.1 User

2.2 System

3 Pre-Conditions

3.1 The user must be register user

3.2 Website must be online

4 Normal Flow

Table

Description automatically generated with medium confidence

5 Alternate Flows

Table

Description automatically generated

6 Sub flows

Table

Description automatically generated

7 Key Scenarios

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# Use-Case: < post article, story or Comment>

Graphical user interface, text, table

Description automatically generated

5 Key Scenarios

Text, letter

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

4 Normal Flow

Table

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# Mental Health Management system (Naaman) private page (online consultation booking)

Diagram

Description automatically generated

# Use-Case: <Make Private Appointment>

1 Brief Description

This use case allows to access the Remote Mental Health Management system to make private appointment for the user. In this stage user can make appointment for meeting the specialist related to their problem type.

2 Actors

2.1 User

2.2 System

3 Pre-Conditions

3.1 The user must be a registered user

3.2 Website must be online

Table

Description automatically generated

Table

Description automatically generated

Text

Description automatically generated with medium confidence

Graphical user interface, text, application, table

Description automatically generated

Text

Description automatically generated with medium confidence

Graphical user interface, text, application, email

Description automatically generated

Text, table

Description automatically generated

7.3 UI and UX

7.4 Language support

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedText, letter

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated

Users must rely on the system that the help option provides immediate response from the above navigation link because it can be urgent, and users may need to get support from psychiatrics more often if the user interface makes it easy for them to use the system.

Text

Description automatically generated

Maintaining consistency is crucial for a remote mental health web application because the users can be emotionally tired when they use the system. The eye-catching and friendly environment which is consistent through all pages will help calm down the system users' minds.

Text

Description automatically generated

Graphical user interface, text, application, letter, email

Description automatically generated

Since the application is based on the web, all the users ranging from physicians to day-to-day users must have a reliable internet connection in order to continue the processes of the system.

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Back-up systems must be maintained in several remote databases in order to secure data and recover easily.

Graphical user interface, text, application, letter

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence Graphical user interface, text, application, letter, email

Description automatically generated

Graphical user interface, text, application, letter, email

Description automatically generated

Architectural Notebook



Risk Management



Master Test Plan



**Post sharing page (Naaman home page with sample posts)**

Used language is PHP and framework is CodeIgniter 3 also using HTML, CSS, bootstrap library as frontend development. This combination is the most common combination in industrial development. Therefore, the development team select these tools and technology for the developing this selected project. CodeIgniter 3 is the most popular PHP framework because it is lightweight in that case if developing php application with CodeIgniter it will run smooth and function without slowing in low performance server it will be benefit for startup project. CodeIgniter run with MVC architecture.

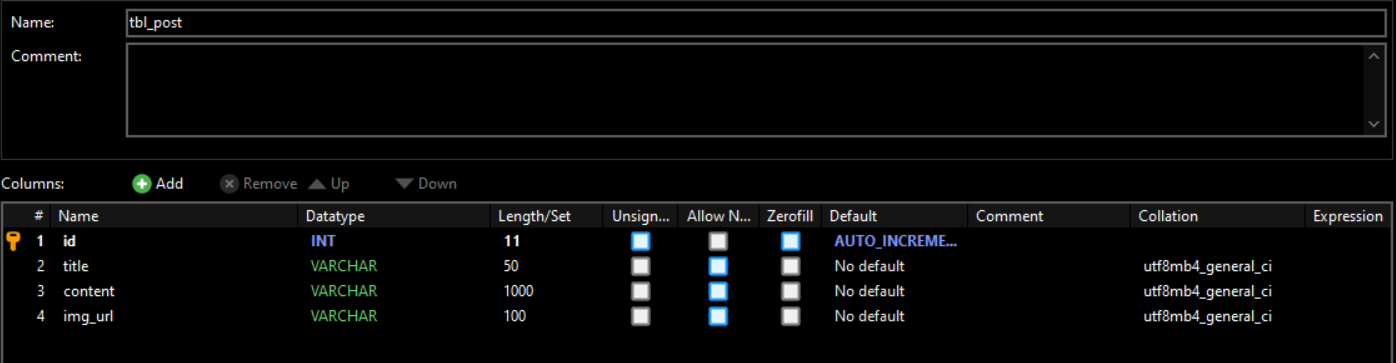
The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects.

CodeIgniter is designed to deliver maximum performance in less time within a clean environment. To achieve this, each developing process is designed in a simplified way.

From technical point of view, it is dynamically instantiation (libraries are loaded on request which makes it light-weighted), loose coupling. (Components rely very less on each other) and component individuality (each class and its functions are narrowly focused only towards their purpose).

This post loads from database

**Database table for using post storing**



**Functioning user interface**

Graphical user interface, application, website

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Test topic 1, test topic 2 and 3,4 are the sample post in database. that posts load in web browser dynamically while the page is loading. Profile pictures and other images must be load dynamically because this content depend on login user. At the same time user can make post or comment using this interface and make appointment.

**Database connection**

Text

Description automatically generated

This is the main database connection in CodeIgniter. Current hosted server is local host (XAMPP). MySQL is the default database driver. Database function is shown below.

**Graphical user interface, application

Description automatically generated**

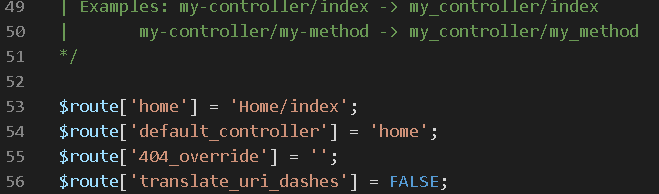
**Base URL configuration code lines (application/config/config.php)Text

Description automatically generated**

Typically, this will be CodeIgniter base URL.

**URL routing configurations**

application/config/routes.php



This route indicates which controller class should be loaded if the URI contains no data.

**View code lines**

Application/view/template/header

Text

Description automatically generated

This is the application view header for using commonly in the application

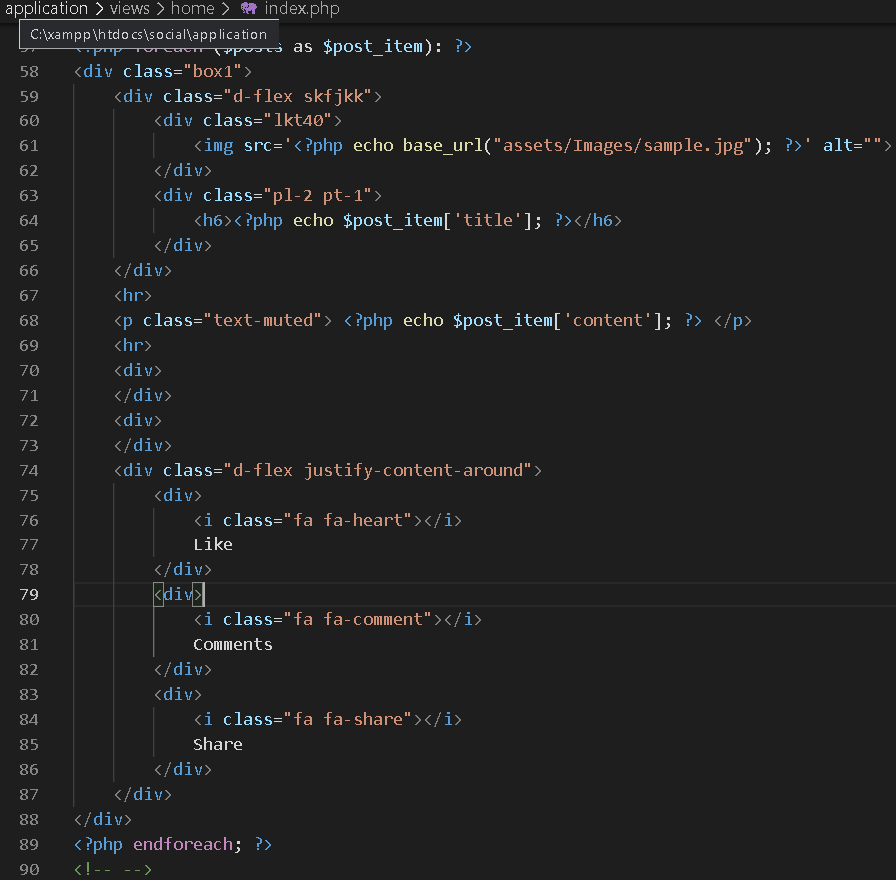
Application/views/home/index.php

Text

Description automatically generated

This is the main home page in application

**Dynamic post loading code lines** (Application/views/home/index.php)



This code line is used to load posts from the database to the user interface. Data sent by the model and past the controller and finally showing the view.

Path is Model > Controller > View

**Controller code lines**

**Text

Description automatically generated**

Controllers control the application function and connect view and model.

**Models code lines**

Text

Description automatically generated

Models is the database connection layer. Models connect to the database and get data to the system and pass the data to the controller. In CodeIgniter has query language for dealing with database it working as MSSQL query.

**User Signup Page (Naaman registration page with sample posts)**

**Functioning user interface**

Graphical user interface, application

Description automatically generated

This is the user registration page that allows user to sign up with the system. It collects user first name, last name, email and password. Its password must contain at least 3 characters.

This UI has designed ensuring the responsum’s that allows user to use the website from using any of their devices.

**UI Code Text

Description automatically generated**

**Controller code lines**Text

Description automatically generatedUser data fields are validated and saving into the database are happening here. Then, also invalid data parameters are handled, and errors are returning to the user. User password is saved as a password hash to ensure the maximum system security.

**User Account Model**

Graphical user interface, text

Description automatically generated with medium confidence

This class represents the user account model. It contains all the user data fields. CodeIgniter base model class has been extended.

**Routes**

Text

Description automatically generated

**Unit Testing** Text

Description automatically generated

PHP unit test has added totest the user signup.

# 

# Registration page and Login page

Here I developed, registration page and login page using Codeigniter 3, a famous PHP framework. Moreover. HTML, CSS, and Bootstrap 4.7 is used to develop the frontend. Our team chose this selection since Codeigniter is lightweight and it is most commonly used in industry. It follows MVC architecture and performs functions smoothly without unnecessary delay.

The Model-View-Controler (MVC) is the common industry standard which is used to develop websites with scalable features integrated. It consists of three main logical components

* **Model**  - Logic regarding data is handled.
* **View**  - Shows the user the information from the model.
* **Controler** - Manages the flow of data into a model object and refreshes the display as data changes.

Since we need to deal with sensitive details in the login page and registration page, for security purposes I added md5 encryption.

**Database table**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**Functioning User Interface**

**Login Page**

**A picture containing text, electronics, indoor, computer

Description automatically generated**

**Registration Page**

**A picture containing text, monitor, indoor, electronics

Description automatically generated**

**Database connection**

Text

Description automatically generated

**Login page**

Here, PHP is used to connect with the database. “md5” encryption method is used to protect sensitive information. The data validations are added. The users can use this page to log into the system.

* **Backend code**

**Text

Description automatically generated**

* **Frontend code**

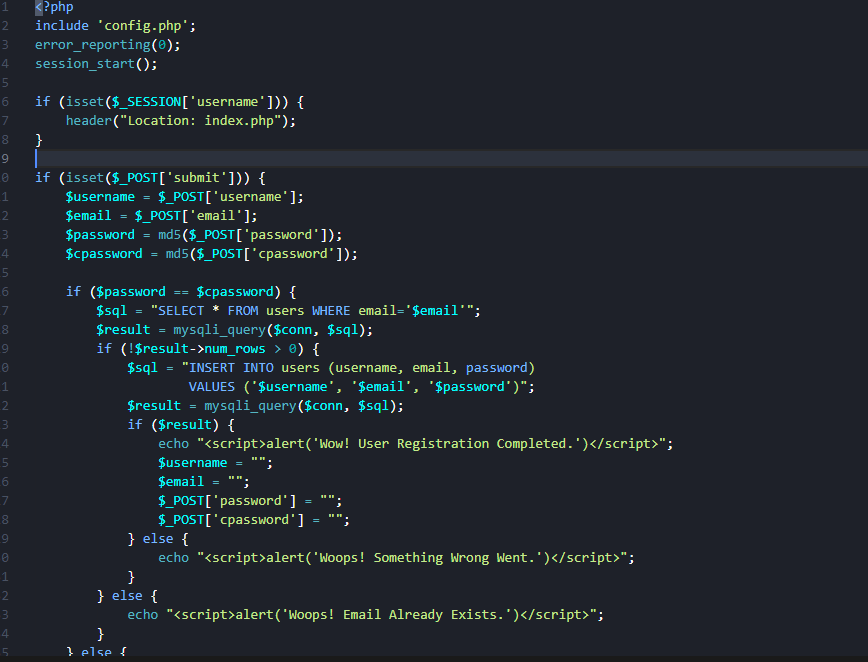
**Text

Description automatically generated**

**Registration page**

Here, PHP is used to connect with the database. “md5” encryption method is used to protect sensitive information. The data validations are added. The users can use this page to register into the system.

* **Backend code**

****

* **Frontend code**

**Text

Description automatically generated**

**Logout**

All the login sessions are destroyed here. The users can use this page to log out from the system.

Graphical user interface, text

Description automatically generated

# Internal User Acceptance Testing (Alpha)

## Scenario Description

* To test the base functionality of the executable architecture
* Test scenarios are a set of tests to validate the executable architecture ability to perform the CCRD use case
* Internal user acceptance testing is performed by the developer to check the functionality of the current state of the product

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date | Author | Description |
| 0.1 | 29/09/2021 | Michael Weisang | Initial Draft |
| 1.0 | 11/10/2021 | Michael Weisang | Initial Version |

## Test Scripts

The following scripts will cover this scenario:

* 1.1 Test Script\_access the web



* 1.2 Test Script\_Registration



* 1.3 Test Script\_login



* 1.4 Test Script\_Posting



## Use Case

* Access the web
* Register new user
* User login
* Post article, story, or comment

## User Groups

* Group 2 “Runtime Terror” Script 1: Access the web

### Script Description

* This test script testing the ability of the user to access the website

### Testing Requirements

This test script covers the following specific testing requirements:

* The website is online
* The user has never visited the website

### Setup

* none

### Teardown

* Document the test result

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Users request the Remote Mental Health Management system over the web browser using link | The browser displays the website |  |

### Test Execution

|  |  |  |  |
| --- | --- | --- | --- |
| Date/Time | Tester | Test Phase | Status |
| 11/10/21 13:35 | Michael Weisang | Step 1 | Failed |
| 11/10/21 15:30 | Dehemi | Step 1 | Success |

* Script 2: Registration

### Script Description

* This test script testing the ability of system to create an account from user inputted data

### Testing Requirements

This test script covers the following specific testing requirements:

* The website is online
* The user has never made an account

### Setup

* none

### Teardown

* Document the test result

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Users request the Remote Mental Health Management system over the web browser using link | The browser displays the website |  |
| 2 | Users go to registration page | The system displays registration page with form |  |
| 3 | Users fills the form and submit it | The system validates the data and record it to the database |  |

### Test Execution

|  |  |  |  |
| --- | --- | --- | --- |
| Date/Time | Tester | Test Phase | Status |
| 9/10/21 13:35 | Sobana | Step 3 | Success |

* Script 3: Login

### Script Description

* This test script testing the ability of system to create a login session

### Testing Requirements

This test script covers the following specific testing requirements:

* The website is online
* The user has an account

### Setup

* none

### Teardown

* Document the test result

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Users request the Remote Mental Health Management system over the web browser using link | The browser displays the website |  |
| 2 | Users go to login page | The system displays login page with form |  |
| 3 | Users fills the login detail and login | The system validates the data and record the sessions |  |

### Test Execution

|  |  |  |  |
| --- | --- | --- | --- |
| Date/Time | Tester | Test Phase | Status |
| 9/10/21 13:35 | Gaury | Step 3 | Success |

* Script 4: Posting

### Script Description

* This test script testing the ability of system to post a story to the homepage

### Testing Requirements

This test script covers the following specific testing requirements:

* The website is online
* The user has an account

### Setup

* none

### Teardown

* Document the test result

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Users request to post the story | The system validates the login sessions  The system expands the post story section |  |
| 2 | Users input the story and post | The system records the data to the database  The system displays the new post in the homepage |  |

### Test Execution

|  |  |  |  |
| --- | --- | --- | --- |
| Date/Time | Tester | Test Phase | Status |
| 9/10/21 13:35 | Dehemi | Step 3 | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Scenario | Tester | Result | Comment |
| 1 | Access the web | Michael | Fail | The reason of failure is still unknown  It speculated be the laptop problem |
|  |  | Dehemi | Passed |  |
| 2 | Registration | Sobana | Passed |  |
| 3 | Login | Gaury | Passed |  |
| 4 | Posting | Dehemi | Passed |  |

Feedback

|  |  |  |
| --- | --- | --- |
| Category  (Aesthetic/function) | Application/system component | Comments |
| Aesthetic | Homepage | The homepage is nice and elegant with a comfortable feeling but the navigation bar at the top of the page looks empty |
| Function | Homepage | Write an article supposed to be a function available for physicians only |
| Aesthetic | Registration page | There is no design yet for the page |
| Function | Registration page | There is a lack of detail still needed in the form to create an account |
| Aesthetic | Login Page | The login page is nicely design and aesthetic |
| Function | Integration of the unit | The integration between the page has met an error |

# Elaboration Phase Status Assessment

## 1. Assessment against Objectives of the Elaboration Phase

### 1.1 Has ‘end-to-end production level support for the most critical, core (risky, difficult) use case, using the chosen software architecture, in the intended production environment’ been achieved?

*Yes,* we *have partially* achieved this objective. This is demonstrated in the online class via presentation.

During the Inception Phase, we identified *posting stories* as the critical core use case. This is because *Posting stories is the main core function of the website.*

We identified *Three-tier architecture* as a feasible approach to addressing the requirements of the projects as outlined in the updated and continuing Architectural Notebook, which may be accessed here:

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1194042_1/A3%20revised%20Architectural%20notebook.docx>

The main architectural elements which are demonstrated by the executable architecture are:

*Registration*

*Login*

*Posting a story*

*View a story*

Those aspects of the architecture not addressed include:

*Posting a comment*

*Posting an article*

*Booking a consultation with a physicians*

Correct support for the CCRD use case by the executable architecture *was partially* achieved as demonstrated and documented in the following user acceptance tests.

*Registration*

*Login*

*Posting a story*

Actual test results can be accessed from here: *<insert link to completed user acceptance tests here>*

### 1.2 Have all critical and significant project risks been addressed and mitigated?

The following list identifies the most critical and significant product, technical and project management risks to the project. Mitigation strategies identified and applied and the current status of the risk are also listed.

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1194187_1/Risk%20Management%20%281%29.xlsx>

### 1.3 Have the initial Vision, Requirements (Scope), or Architecture changed?

During the Elaboration Phase, our understanding of the projects aims evolved as follows:

Vision

*There is no major change in the vision document. There are only a further, more detailed explanation or description added to the vision document. The reason for the added detail is because of the brief information provided is not enough.*

Requirements (Functional)

*There is no major change in the Use case description as in the initial use case, the CCRD use case is already mentioned. The change happening in the document is only a further detail added to the current use case so that it will be more specific.*

Requirements (Non-Functional)

*There are no major change in the Non-Functional Requirement documents. There is only a minor added information to further explain the NFR requirement.*

Architecture

*There are no major changes in the architectural style. There are some minor detail added for the architectural notebook to be more specific and detailed.*

### 1.4 Have the initial Project Plan or Master Test Plan changed?

During the Elaboration Phase, our understanding of the best way to implement the project evolved as follows:

Project Plan

*The project plan is not changed at all*

Master Test Plan

*There is a major change happening in the master test plan. One of the reasons of the massive change is that the initial master test plan are not in the best state with a lot of information lacking. In the revised master test plan, a lot more detail has been put and some of the information like planning are changed.*

## 2. Deliverables

*For each deliverable:*

Text

Description automatically generated

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1193475_1/A3_Vision%20Document.docx>

No Issue

Text

Description automatically generated

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1194042_1/A3%20revised%20Architectural%20notebook.docx>

No Issue

Graphical user interface, text

Description automatically generated

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1194187_1/Risk%20Management%20%281%29.xlsx>

No Issue

Text

Description automatically generated

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1193981_1/revised%20Use%20Case%20Description.docx>

No Issue

Text

Description automatically generated

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1194045_1/NFR.docx>

No Issue

Graphical user interface, text, application

Description automatically generated

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1194189_1/A3_Master%20Test%20Plan.doc>

No issue

Text

Description automatically generated

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1174030_1/Project%20Plan.docx>

No Issues

## 2.8 Domain Class Diagram

## https://interact2.csu.edu.au/courses/1/S-ITC303\_202160\_SM\_I/groups/\_68438\_1//\_1194212\_1/Domain%20Class%20Model%20Diagram.PNG

## 2.9 ERD Diagram

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1194211_1/ERD%20Diagram.PNG>

## 2.10 Homepage and sharing document

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1193982_1/Home%20page%20with%20post%20sharing.docx>

## 2.11 Registration document

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1194192_1/User%20Signup%20Page.docx>

## 2.12 Login Diagram

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1194208_1/Log%20in.docx>

## 2.13 User Acceptance Test

<https://interact2.csu.edu.au/courses/1/S-ITC303_202160_SM_I/groups/_68438_1//_1194209_1/UAT%20Test%20Scenario%20and%20result.docx>

## 2.14 Executable Architecture

<https://github.com/Temuulen2830/ITC303-Remote-Health-Management/tree/Dehemi>

<https://github.com/Temuulen2830/ITC303-Remote-Health-Management/tree/Sobana>

<https://github.com/Temuulen2830/ITC303-Remote-Health-Management/tree/main/Login_registration>

Major Issue in codding. The developer delivers the code late and did not use the specification specified in the project. This has led to no time for integration

## 3. General Issues

*For each issue*

### 3.1 Time

*Due to long due date in the assignment, the team member keeps on delaying the project until there is not enough time to properly finish the deliverables.*

*The issue is still ongoing and for the next phase, there will be a stricter monitoring*

### 3.2 Coding

*Due to delaying the task assigned, the team member ends up rushing the coding near the end of the term which led to major issue in the end.*

*The issue is still ongoing and for the next phase, there will be a increased pressure to finish on schedule.*

## 4 Summary – Overall Project Progress

*The project is progressing albeit slowly. Most of the aim for the elaboration phase has been fulfilled except the executable architecture. The unit is developed nicely, but the integration has major problem. With the current pace, the project is facing risk of failure if there is no major change taken by the team. Thankfully in the next phase (Construction phase), all the major issue faced in this phase can be fixed and reworked.   
  
There is no change for the project scope and project plan*