

Log book

for software developers working on a e-health system.

The aim of our project, as specified in the criteria, is to develop “an online healthcare appointment management system (HAMS) that would streamline the process of patients booking healthcare appointments and general practitioners sending referrals to specialists. The client feels that such a system would help to market and increase the likelihood of attracting more customers to the health centres affiliated with MediSoft”.

In order to implant this software, a structured work system is required in which we are weighing out all the requirements and working in layers, planning first the objects and the abstract data types of our software on paper, confirming the details with our supervisors in person and proceeding to implement our blueprint into a working program.

Name	Position	Organisation	Phone	Email
Za Hi	A member of the software developers team	UNSW		
Oren Levi	A member of the software developers team	UNSW		
Gian Kongruangkit	A member of the software developers team	UNSW		
Alyona Berkovich	A member of the software developers team	UNSW		

Log book

for software developers working on a e-health system.

Date / period	Experience and decisions made	Requirements Elicited	Task to Complete
week 4	Planning and developing the software.	Begin a detailed outline of the tasks to be allocated and assigned based on strengths and weaknesses.	Detailed outline of tasks.
Week 4 (meeting on 17.8)	Analysing the assessment criteria and the requirements of our software.	Becoming extremely familiar with the tasks required of us.	Meticulously read through the specification sheet.
Week 5 (meeting on 20.8)	Writing user stories.	Identifying the users and focusing on the expectations from the users' roles in our software.	User story description and criteria.
Week 6 (meeting on 27.8)	Writing CRC cards.	Identifying the classes and all the responsibilities they have.	CRC cards.
Week 7 (meeting on 4.9)	Creating a UML diagram.	Inspecting the requirements, we drew our attention to during last meeting.	UML Diagram, first draft for lab review.
Week 7 (meeting on 5.9)	Making sense of and identifying the best way to read in the data given to us in the CSV files.	Working with dictionaries and writing code and discussing more tasks to allocate each other.	Read in data from CSV file.

Log book

for software developers working on a e-health system.

Week 7	Have the code successfully iterating through the CSV files.	Assigning research tasks for pickle and packages to assist file manipulation.	Exploring picking and Python packages
Week 7	Coding the classes and creating other fundamental features for our software. Having all the classes and sub classed ready by next meeting.	Focusing on creating all the skeleton for our four super classes, Appointment, Centre, Health, and User Class.	Create basic class attributes, constructors and methods.
Week 8 (meeting on 10.9)	Finished the attributes and methods needed for all sub classes.	Refine and look over the code written in the last few days to determine satisfaction with all the subclasses created from all parent classes.	Review the classes and create subclasses.
Week 8	Login authentication to start checking for valid accounts.	This was completed earlier though issues arose between patient and provider. Need to resolve the difference in displaying pages to the respective user logging in.	Login to verification.
Week 8	Generate different Login for page for patient and provider.	Have users work on flask URL based on login as patient or provider.	Generate different page based on user login.
Week 8	Discuss the design and implementation of the consultation page.	How best to seperate, display and read the notes and prescription of medicine.	Begin coding consultation page.

Log book

for software developers working on a e-health system.

Week 9 (meeting on 17.9)	Discussing how best to implement the search engine. Breaking up the four services, into Centre, Suburb, Provider, Service. Identifying how similar words need to be in order for search engine to pick up on a match.	Refining the search engine to identify closely related words. Adding the dropdown menu to show the four different services that will each display the centre or provider depending on what is selecting.	Search by Centre, Suburb, Provider, Service.
week 9	Changed our initial decision to have current and past appointments on same page, to now have a seperate tab for current appointments and past appointments, under history.	Decide the best way to take all information from appointments and move to history once the provider has ended the appointment.	Create an individual page for current and past appointments.
Week 9	Search engine to include specific word searches, after selecting one of the above four services.	To write code to pick up on specific centres and providers entered by the user. Successfully added the similarity of misspelled word to show related strings.	Search by specific word.
Week 9	Experienced lots of merge conflicts.	Go through after Monday lab to identify the old and new pieces of code we need to keep and abandon.	Resolve merge conflicts.
Week 10	Debug appointment and history errors.	Go through debug appointment and past appointment display errors.	Debug errors.
Week 10 (meeting on 27.9)	Looking at python packages like "Pylint" to assist in cleaning up code, identify hidden bugs, and recommend proper use of syntax.	Download and run Pylint in order for each member to use on our allocated parts of the code.	Test Pylint on code.

Log book

for software developers working on a e-health system.

Week 10	Discuss the aesthetic appeal of the website. How we want the site to look, the colours, banners, words and all.	Avoid using symbols or logos or brands that associate our company closely with others. Try be generic and individualistic.	Design unique aesthetic appeal.
Week 11 (meeting on 5.10)	Begin running test cases for the website. Looking at edge cases and navigating through each and every possible avenue to try find bugs.	Have each member go through the website looking for any errors in functions and/or features.	Run tests on code.
Week 11	Centre name seems to be coming up as individual.	Big bug needs to fixed. All eyes are looking for the problem so we can generate a solution.	Resolve centre name problem.
week 11	Write text files containing all the problems found in our website, namely the issue with submitting rating taking user to a new page.	Go through and begin systematically resolving each problem based on importance.	Identify and resolve each problem found.