

Authors

Johan Andersson
David Fogelberg
Sam Halali
Gunnar Gunnarsson
Nandha Gopal Elangovan
Miguel Angel Sanchez Cifo
Jonathan Granström

Git repository: https://github.com/lol2kpe/EDA397_Team3
Issue tracker: https://github.com/lol2kpe/EDA397_Team3/projects/1

Project Description

Project Title

H4U

Platform

Android

Minimum and target SDK

5.0

Description

The H4U app helps a user search for hospitals, pharmacies and doctors including; doctor details, doctor availability and doctor reputation. It will also show the directions to each hospital by showing the route map. The route map will be set from a user's current location and will show both a visual and descriptive way of the route, which will make it easy for a user to find the hospital location.

Sprint Log: Sprint 1

Commitment

List the features/stories the team commits to finish during the sprint.

User Stories

ID	User story
2	As a user, I want to be able to search for doctors
3	As a user, I want to be able to search for hospitals
4	As a user, I want to be able to make an emergency call
5	As a user, I want to be able to search for pharmacies
7	As a user, I want to be able to create my own profile
8	As a user, I want to be able to get directions to a hospital
9	As a user, I want to be able to schedule appointments
10	As a user, I want to be able to get directions to a pharmacy
11	As a user, I want to be able to consult with a doctor

ID	User story
12	As a user, I want to be able to see all the nearest hospitals in my area
13	As a user, I want to be able to update my profile
14	As a user, I want to be able to set medicine reminders

Tasks

ID	Tasks
41	Add/remove markers from map
38	Save user's filter selections
37	Add basic filter options
36	Create navigation drawer
35	Create functional filter functionality
32	Create Doctor object
31	Create Pharmacy object
30	Create Hospital object
28	Add map to main activity
25	Display directions on the map
24	Fetch directions based on users location for selected hospital
23	Add function for Sending requests to database
20	Add filter option to GUI
19	Fetch from database (create object from database entry)
18	Connect to database
29	Create markers for map
27	Create database contents
26	Create/start database server
17	Find user's location
16	Initiate map with Google API
42	Create dummy data for map

Work Done

Feature	Commits	Group members	Effort	Practices
Find user's location	66dfdfa	Sam Halali Miguel Angel Sanchez Cifo	2h	Pair programming
Create/start database server	c61f98fd9f984ef36f379	Johan Andersson	7h	Simple Design, Continuous Integration
Seed database with objects	31c7c57	Johan Andersson David Fogelberg	35m	Pair programming
Create Doctor object	ff25688	David Fogelberg	1h	Test-first

Feature	Commits	Group members	Effort	Practices
Create Pharmacy object	ff25688	David Fogelberg	1h	Test-first
Create Hospital object	ff25688	David Fogelberg	3h	Test-first
Create markers for map	ba9e5da	Sam Halali	2h	-
Add map to main activity	b003734	Miguel Angel Sanchez Cifo Gunnar Gunnarsson	1h	-
Create database contents	5ee2fa7	Miguel Angel Sanchez Cifo Nandha Gopal Elangovan	4h	-
Fetch from database (create object from database entry)	3483f4f	Johan Andersson	3h	-
Connect to database	3483f4f	Johan Andersson David Fogelberg	3h	Pair programming
Find user's location	66dfdfa	Sam Halali	3h	-
Initiate map with Google API	b003734	Miguel Angel Sanchez Cifo Gunnar Gunnarsson	1h	-
[Save user's filter selection]	2564636	Jonathan Granström	2h	Testing first, Simple design, refactoring
[Add basic filter functionality]	aa31251	Jonathan Granström	2h	Simple design
[Create functional filter functionality]	72f4557	Jonathan Granström David Fogelberg	5h	Refactoring, simple design, pair programming
[Add/remove markers from map]	c042c79	Gunnar Gunnarsson Jonathan Granström David Fogelberg Miguel Angel Sanchez Cifo Sam Halali	1h	Pair programming, simple design
[Report]	[60ed7d4]	Nandha Gopal Elangovan	2h	Small release
[Create dummy data for map]	[46cb5fe]	David Fogelberg	1h	-

Reflections

Brainstorming was used to gather ideas for the project proposal. After getting the proposal accepted, the same method was used to gather ideas for functionality, these ideas were then formulated into user stories. Another

brainstorming session was performed to decompose the most important user stories into tasks.

The 100\$ method was used to prioritize the user stories according to relevance to the applications goals. It was interesting to see how unaligned our estimates were except for those that clearly contributed to the goal.

Pair programming was used to solve challenging tasks and worked really well in the group, both in the start of the sprint and the end of the sprint. It was used to solve for example fetching data from the database, integrating data with filter and integrating the map with the rest of the functionality.

Testing worked quite good for specific methods however for Android GUI parts it was more challenging. When it came to test-first it was really good when you got used to it but it is also something that we need to work on the next sprint. Sometimes you started writing the functions without writing tests first because you are so used to that kind of programming.

Refactoring was used to improve the code and make it more readable. It is really good to always consider refactoring the code since there is almost always something to improve.

We definitely would have liked more time for the planning phase of the project. Within the short span of time that was given from the group formation to the delivery of Sprint 1 only gave enough time to gather the most important tasks and delegate them before starting to implement them. Methods we would have liked to use during the planning phase in the next sprint:

- Analyze dependencies
- Planning poker to estimate work
- Software architectural design