Psyche Assistant

Project Definition

Extended functionality for mental health oriented app.

Supervisor: Flemming Sørensen, Teacher

Consultant: Simone B S Larsen, Occupational Therapist

Developer: Mads Søndergaard, Student

Contents

Foreword:

In this document, I will first introduce the reason why this project is deemed worthwhile. This then leads into the concept of how to solve the matter at hand.

I will then proffer a timetable for what each leg of the project will consists of practically, and what chronological order I assume it will take place in.

Subsequently, the Software Requirement Specification proper will be introduced, detailing the software's purpose, the current situation in terms of what is offered and what is needed, the criteria for meeting those demands, followed by an analysis of the information required to accomplish that task.
I will then go over the profile of users of the app, as well as the process of using the (*extended*) functionality of the application, along with what prototyping is deemed necessary to ensure the functionality that is being developed adheres to the goal of meeting aforementioned criteria.

This is then followed by the software requirements proper; the models to be employed, their relation, the UI/UX to be developed and the functions that are deemed necessary to accomplish the goal of the functionality to be implemented.
At the end of this section, I will briefly summarize what options are readily apparent for further extension of the app, as well as what platforms/tools are to be employed.

The software section will then be followed by a hardware section, which details the devices used during the development and/or for the functioning of the application, to the extent that it is known.

Penultimately, I will give a brief summary of the need-to-have and nice-to-have functionalities talked about during this document, as well as how to test these, and what security measures are deemed necessary.

And finally I will list some of the nomenclature and/or references used, if applicable, as well as an overview of the documents history, for the sake of transparency.

1. Problem Definition:

After the relatively recent COVID epidemic, it's become apparent – especially for those afflicted and those who tend to work closely with said afflicted – that mental fatigue is a prevalent side-effect following the infection.

This in turn brought into light the prevalence of the affliction in other scenarios, and has since been found to be prevalent among not just those who have been inflicted by COVID19 at one point or another, but also patients who have had other moderate- to severe infections, or have suffered brain trauma, or undergone emotional- or psychological trauma.

Unfortunately, mental fatigue is often overlooked, either entirely or as a secondary effect, and it thus continues to avoid the general consciousness, including both patients and professionals, meaning there is a substantial obstacle both for patients to receive adequate care, and for professionals in the tools at their disposal.

Developed by the author, Mads Søndergaard, in collaboration with the Subject Matter Expert Simone B S Larsen, the app Psyche Assistant acted as the proof of concept for an app that could function both as a self-management tool for patients, and diagnosis-tool for professionals, allowing for the (*anonymous*, *non-registered*) user to take a survey with hard-coded questions.
Based on how they scored the questions posited in that survey, the app calculated a score reflecting the users level of Mental Fatigue, and based on that score, presented generalized recommendations and symptoms for that level of Mental Fatigue, which in turn were then stored locally on the users device, generating an overview of results over time.

The goal of this project, then, is to investigate (1) how to extend the app to be more of management tool, as opposed to the current focus on diagnosis. (2) Implementing functionality that would allow users (and those close to the user, so to speak), to try and manage their energy expenditure rather than simply diagnosing it. (3) Doing so in a manner that lets the user(s) see their daily energy expenditure, by way of using activities assigned an energy-cost, for example, or at the least giving the user(s) tools to help patients cope with their mental fatigue.

If time permits, the project would also like to investigate whether it is possible to extend the current diagnosis-aspect of the app, and how it could be made more dynamic, allowing for the rolling implementation of other diagnostic survey tools.

2. Concept:

The primary goal of this project will then be to extend the app with functionality that further deepens the tools available to the patient (and their relatives), so they can manage and track their activity levels based on the activities they undertake, and users in the same group can see these results, allowing for those close to the patient to partake in management of the patients brain fatigue, by for example seeing group-based activities and acting upon them, and/or see the energy expenditure of other members in the group.

This would require adding a wholly novel interface for:

1. User and group registration/assignation.
2. The creation and management of "Activities", and whether these contain tasks, are defined for the user alone or group, etc.
3. An overview of these activities, preferably on a daily basis, so as to avoid overexposure for the patient, potentially exacerbating mental fatigue as a result.

For the above to be feasible, an external database would be required to store users and activities, as well as their super-relations (groups) and sub-relations (tasks).

If time permits, this would then also allow for parts of the core components of the app Psyche Assistant to be rewritten for take advantage of this extended functionality.

Specifically:

- Rewrite the storage logic for surveys so it targets an external database rather than a local one.

- Rewrite surveys so they are assigned a specific user and can thus be fetched.

- Rewrite the interface so it supports a more dynamic approach to surveys, rather than the reliance on hard-coded JSON/resource files, after which new surveys could feasibly be generated on-the-fly, and published for users to access via drop-down.

Users could then see an overview of their results on various surveys, based on the type of surveys available (e.g. "mental fatigue" *(Mental Fatigue Scale*), "depression" (*Beck Depression Inventory*), "sleep" (*Pittsburgh Sleep Quality Index*) etc.), allowing for the user to track and manage more aspects of their mental health than just mental fatigue.

This would in turn create a more generalized overview of the patients general mental health, and potentially function as a tool to identify primary and/or secondary diagnosis, as well as streamline tracking and management guidance for professionals.

3. Project Plan:

Development days total: 11 **Documentation days total: 4,5**

Day 1 – 2:

Design the basic skeleton for models and back-end for users and activities, determining a fundamental model design for the necessary objects to be used.

Ideally, this should result in the object definition of users, groups, activities, and sub-activities.

Day 3 – 4:

Establish an external database that can store the objects mentioned previously, as well as a way for the app to communicate with the database and CRUD said objects to the extent needed.

Ideally this should allow for the exchange of objects between the app and database, laying the grounds for UI/UX implementation.

Day 5 – 7:

Design the additional front-end UI/UX for User registration and login, as well as the logic needed to support this functionality, as well as the creation and opt in to specific groups.

Ideally, this should allow for the creation and logging in of specific users, and provide a method for the user to join a specific group.

Day 8 – 10:

Design the additional front-end UI/UX for Activity/Sub-activity creation and management, as well as the energy expenditure related to these.

Ideally, this should then allow for registered users to create activities, and the other members of their group to see these activities.

Day 11:

Polish and functional testing to try and weed out potential bugs.

Day 12 – 16:

Writing of reports.

4. Software Requirement Specification

4.1. Purpose:

To extend the app "Psyche Assistant", which currently only offers diagnostic tools using hard-coded survey and feedback data using a local storage, with a management tool that allows the user to register and create/join a group, and create activities, optionally with sub-activities, that the group can see and act upon, if necessary. As the mark an activity as completed, the activity's energy-cost is added to the users daily energy expenditure, allowing the user(s) to track how much energy a given user has used up on a given day.

The intended use is then primarily the patient and those who are in a position to support them, such as their family, for example.

It is to be developed by Mads Søndergaard, who also developed the foundation for the aforementioned app, with use case feedback and subject matter expertise provided by Simone Larsen (occupational therapist).

4.2. Current situation:

The app currently provides a rudimentary tool for periodical assessment of a persons level of mental fatigue, using the official Mental Fatigue Scale survey. This in itself is a valuable tool for meta-tracking the effect of daily life on the users mental energy stores. However, it is limited in its granularity as well as scope, meaning it doesn't directly indicate what might have caused the mental fatigue to increase or decrease, only that it has, and the responsibility of using the app still rests solely on the patient themselves (or their occupational therapist; whoever is adamant enough to insist they take the survey), which can be very challenging, especially for the moderate- to severe cases of mental fatigue.

The intended solution, then, is to create a tool that allows a more granular resolution of what activities might explain the increase or decrease, by allowing the user to create activities with a proffered energy cost, and complete these, thus giving them a tool to both see how much energy they spend during a day, but also whether the concept they hold for a given activities energy-cost reflects the real-world energy-expenditure, based on the activities carried out, their assigned score, and the impact on mental fatigue.

4.3. Criteria:

For the software to be considered successful, it must (1) provide the user with a tool with which they can feasibly attain more granular knowledge of their energy-expenditure, and activities energy-cost, (2) provide the user with a tool that enables them and others to better manage their mental fatigue by way of sharing the workload of various activities.

4.4. Information analysis:

The extended functionality will consist of Users, Groups, and Activities, where a Group can have more Users, and Users (*initially, at least*) can be a member of 1 group, with the potential for one or more Activities to be assigned to the group. When a user carries out the activity and completes it, that activity's energy cost is added to the users energy expenditure for the day.

4.5. User Profile:

The user(s) of the app will first and foremost be (1) an individual suffering from mental fatigue (or other mental challenges that may limit their ability to function to some extent), who could benefit from (2) others – relatives, friends, etc. – that could assists them in that function, or who need to be aware of their energy expenditure, if nothing else.

To give a theoretical example:
A person suffering from mental fatigue, unaware of their condition, searches for their symptoms, and stumble upon this projects app.

They install it, and are greeted with a simple landing page, initially empty.

They can then undergo the preexisting survey, acting as a preliminary diagnosis tool for mental fatigue; they find that their symptoms fit with severe to moderate mental fatigue.

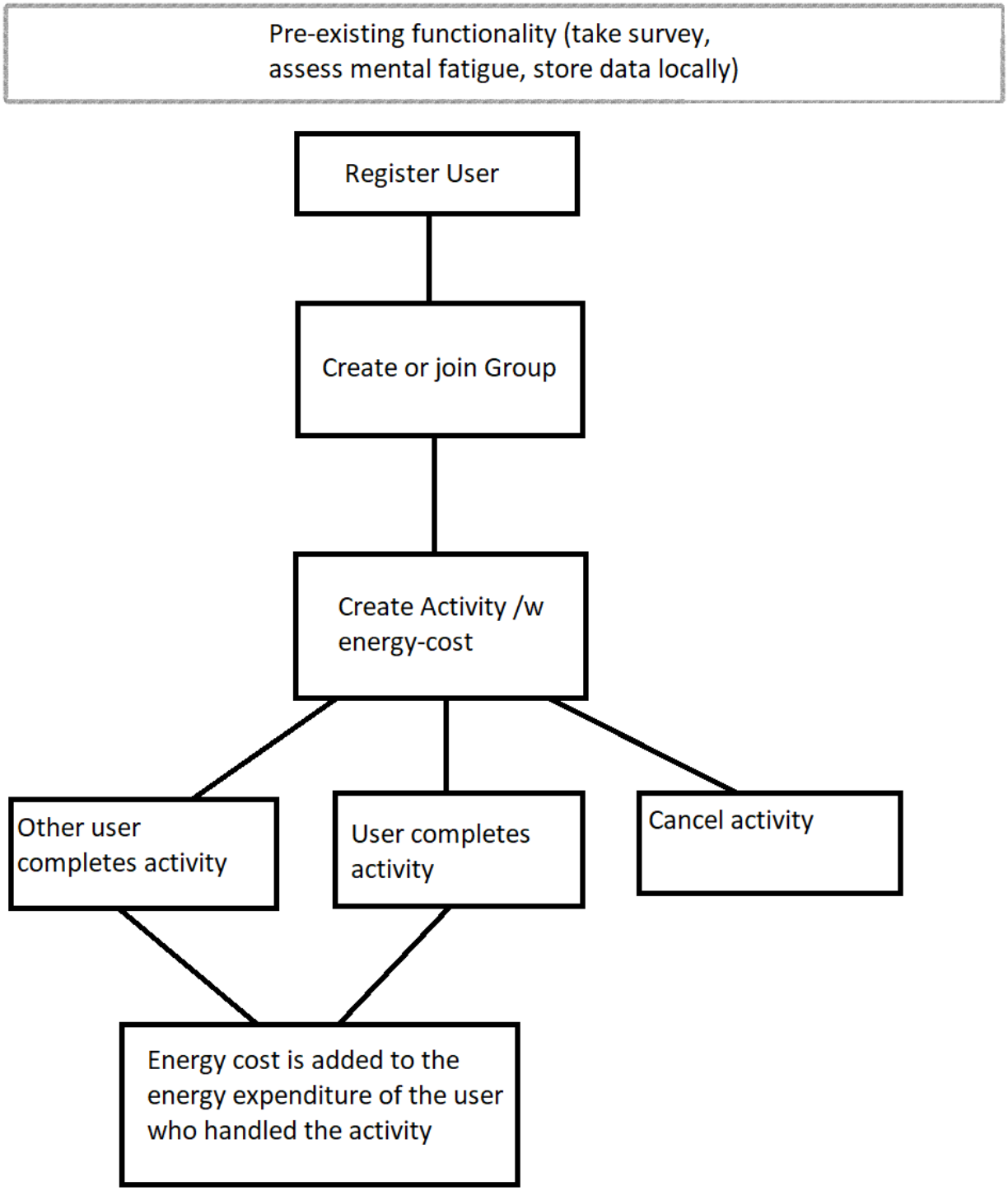
They notice the Activity tracker, and try to access it, but since they aren't logged in, they are forwarded to the registration page, allowing them to create a user.

They can then create or join a group, for example based on an invite, be it a link or a simple randomly generated key that allows them to connect their user to a group.

Finally, they can then create activities and estimate their energy cost, and when a user then marks that activity as complete, the energy cost associated with it, is added to the users energy expenditure.

This would then update a group overview of the groups members, allowing everyone to follow each members energy-expenditure for the day.

4.6. Process:


Pre-existing functionality (take survey, assess mental fatigue, store data locally)
Register User
Create or join Group
Create Activity /w energy-cost
Other user completes activity
User completes activity
Cancel activity
Energy cost is added to the energy expenditure of the user who handled the activity

4.7. Prototype:

Prototyping will primarily be in terms of verifying the creation of users, groups and activities when appropriate, but beyond that, the existing app ("Psyche Assistant", see references for link to relevant Github commit), could be argued to be a prototype, as it serves as a proof of concept, albeit the functionality it holds as that proof of concept isn't related in a technical sense to any of the functionality this project sets out to add to it, only theoretically in its applicability.

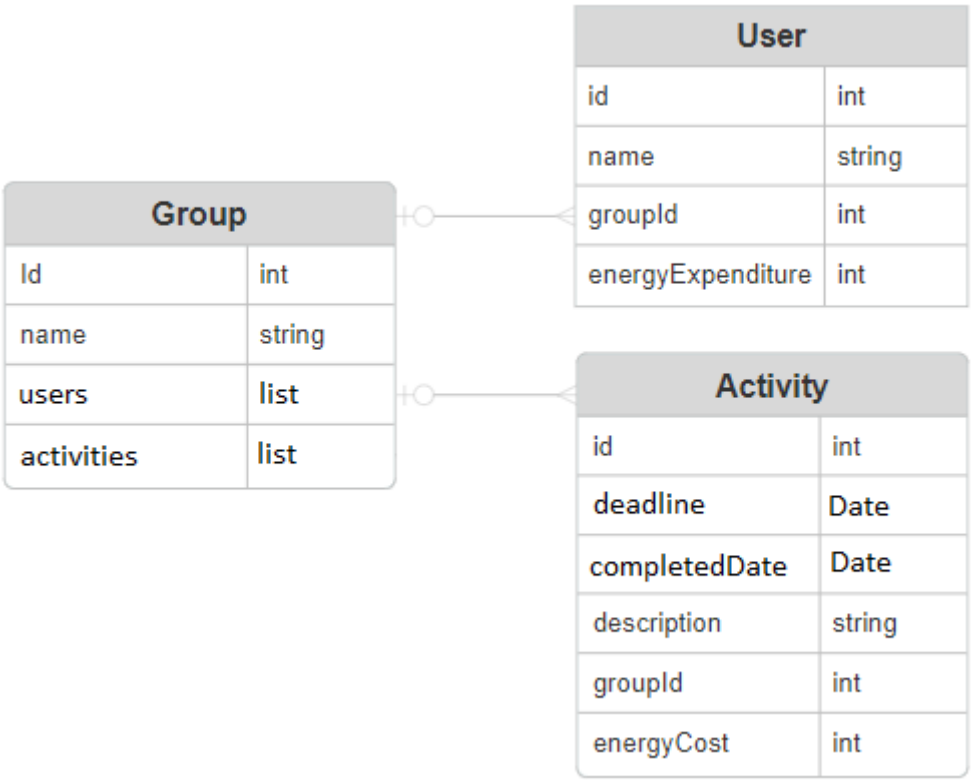
4.8. Software:

The following is a brief overview of the software architecture necessary to accomplish the primary goal of the project.

4.8.1. Models:

The following are the models needed to store and manipulate data relevant to the desired function of the extension to the application.

4.8.1.1. Entity Relation Model:



Note here that both Activity and User models are in a O2M relation with Group models, i.e. a group can have one or more users, and zero or more activities.

4.8.2. UI/UX:

The app currently has a simple layout, where a history of results from taken surveys are shown on the landing page, if relevant, as well as a top navigation bar used for navigating to and from the survey, and a bottom navigation bar used for selecting the language the questions should be fetched in.

This will be extended with:

- Additional menu items in top navbar, with additioanl functionality on each page:
  - Activity page
    - List of user relevant activities (i.e. activities assigned to the group they are part of)
      - Buttons to complete or cancel individual activities.
    - Form for creating a new activity:
      - Deadline
      - Description
      - Energy Cost
      - Button to submit
  - Group page
    - The group name/identifier shown at the top.
    - List of users who are a member of the group in question
      - With their energy expenditure shown
    - Button to Create/Join group if not currently member of one.
    - Button to Leave group if currently member of one.
  - User page
    - Button to log out/delete user.
    - Button to log in/register if app isn't connected to user.

4.8.3. Programmatic Functions:

The following is a list of the general functions that the extension should have – either explicitly or implicitly – to facilitate fundamental functionality. This list doesn't include potential Helper functions, such as a function to add an Activitys energy cost to a Users energy expenditure upon completion.

Name:	**Functionality:**
CreateUser	Creates a new user in the database and assigns it as the apps user
UserLogout	Logs out the currently signed in user
UserLogin	Logs in a user based on the given values, if possible.
ReadUser	Gets information of a specific user
DeleteUser	Deletes the currently logged in user from the database.
CreateGroup	Creates a group in the database
DeleteGroup	Fetches the connected groups name/identifier
ReadGroup	Gets information specific to the group
LeaveGroup	Removes the users connection to the group
CreateActivity	Creates an activity for the users group in the database
CompleteActivity	Marks the activity as completed as of todays date.
DeleteActivity	Removes the activity from the database, if not marked as completed.

4.8.4. Extensibility:

As mentioned previously in Section 2 (Concept), the application could be extended even further, once the new functionality is in place.

For example, one could move the currently hard-coded survey data to the external database necessary for the project primary functionality, and rewrite the proof-of-concept to allow for the dynamic creation of new surveys, offering users a list of potential surveys they could undertake, rather than just the one.
This could in turn benefit form the existence of an Administrator who could vet and publish these surveys, and in turn, each survey taken could be stored with type, score and date, allowing users to follow the progression or regression of whatever mental or psychological affliction the survey is focused on.

Alternatively, focus could be shifted to extend the existing functionality to iOS, in order to make the application more widely available.

4.8.5. Platform:

The main focus is to extend the Android portion of the proof-of-concept application, using Kotlin Multiplatform (or "Compose Multiplatform", as it is sometimes called, when both UI and Logic are shared across platforms).

Contrary to the foundational app, however, the focus here is to store the data externally, off the users device, as this is necessitated by the desired group-functionality.

This will most likely take the form of an SQL database, run either on a fitting open-source cloud platform if available, or for the sake of the examination, on a self-owned miniature server, such as a Raspberry Pi of some sort.

Currently, it is presumed that Spring Boot will be used to serve the API to communicate with the database.

4.9. Hardware:

For development, a Samsung S21+ running Android 13 will be used.

If no other option is found, for the purposes of the examination, a Raspberry Pi will be used to run the external server/database – exact model and so on to be determined, based on funds and/or availability.

4.10. Requirements:

4.10.1. Need-to-have:

- User, Group, and Activity models and CRUD
- External database for storing and retrieving data
- UI/UX for managing users/group/activities

4.10.2. Nice-to-have:

- Move survey to external storage.
- Add CRUD functionality for surveys.
- Refactor UI/UX to allow for dynamic selection/overview of available types of surveys.

4.11. Testing:

1. Can the user create a User?
2. Sign out and sign in?
3. Does deleting the user delete their data?
4. Can the user create or join a Group?
5. Can they leave it?
6. Can a user create an Activity?
7. Does completing the activity affect the users energy expenditure based on the activities cost?
8. Can an activity be deleted after it has been completed?
9. Can other users complete an activity created by another user?

4.12. Security:

Considering users must log in using credentials, it would be essential that their credentials are sufficiently encrypted.

4.13. References and Nomenclature:

- Starting point for Application:
  - https://github.com/MadSantiak/H6/commit/553a5e5095cf92ff3a6fc5ae50a42746b5ca5bc2
- MSF: Mental Fatigue Scale
- BDI: Beck Depression Inventory
- PSQI: Pittsburgh Sleep Quality Index
- UI: User interface
- UX: User experience

5. Document history:

29/7-2024: Draft by Developer (Mads Søndergaard). Checked by Supervisor (Flemming Sørensen).

30/7-2024: Finished draft submitted by Developer for review by Supervisor before hand-in.