Functional Specification Contents

Noted.

Student Name - Mike Purcell
Student Number - 15446908
Supervisor - Jennifer Foster

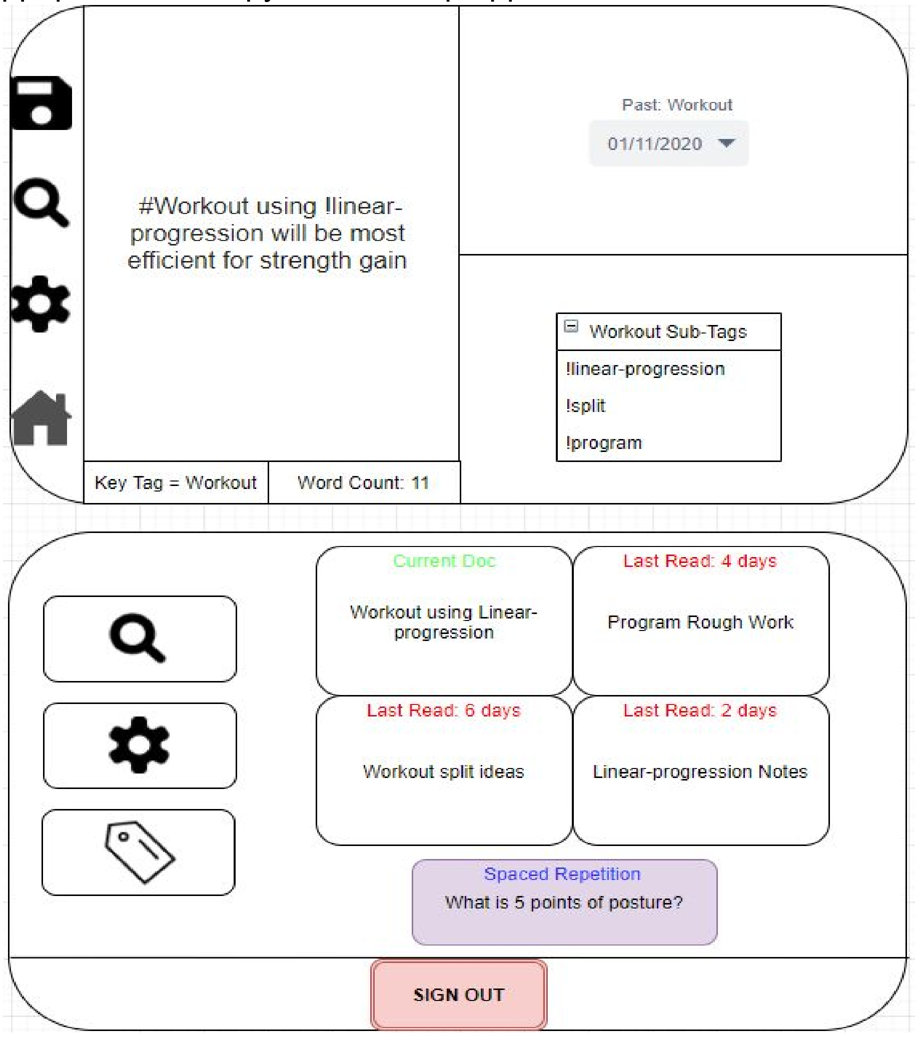
0. Table of contents

1. Introduction

1.1 Overview

The idea is a Note taking rest api with a web app front end.Notes will be stored on a database with my rest framework.
The goal is to have a unique note taking interface to allow past notes to become present dynamically during your writing. This is done through Tags(#) and SubTags(!). Once a Tag is defined using a # all future occurrences will automatically be selected and stored.
As you begin to type and use a stored tag, the app will begin displaying past notes under the same topic. Another core concept of the app is spaced repetition as a learning method. If selected, a note will be saved to then be displayed over intervals of days/weeks to avail of the spaced repetition.

As there are no note taking interfaces that use a similar concept of the dynamic view, I feel it fits into a niche that hasn't been explored yet. It will be a Web App, paired with a python desktop app.



1.2 Glossary

- **Tag/Subtag**: Tag(#) defines a topic, a Subtag(!) specifies within a topic.
- **Spaced Repetition**: Spaced repetition is an evidence-based learning technique that is usually performed over days/weeks/months depending on difficulty of topic.
- **API**: A software intermediary that allows two applications to talk to each other.
- **Containerize**: Application containerization is an OS-level virtualization method used to deploy and run distributed applications without launching an entire virtual machine (VM) for each app. Multiple isolated applications or services run on a single host and access the same OS kernel. Containers work on bare-metal systems, cloud instances and virtual machines, across Linux and select Windows and Mac OSes.
- **Pod:** Pods are the smallest deployable units of computing that you can create and manage
- **MVP**: A minimum viable product

2. General Description

2.1 Product / System Functions

The goal of the app is to allow users to be able to do all the necessary functions of regular note taking apps within one place.

The User will have the ability to select a note type and allow it to be saved and interacted with accordingly. A note can be selected as a finished note, storing it away until dynamically brought up with tags or manually searched for. A note can be selected as in progress and after saving will continue to be shown to the user until it is marked as complete. A note can also be selected for spaced repetition, allowing it to be used for study being shown to the user in a spaced manner over days/weeks.

Analysis of the users notes will be displayed such as total words typed, word occurrence etc.. As well as metrics to evaluate the apps performance for myself using Datadog.(Things like are the suggested notes being interacted with, length of time to click certain buttons etc..)

2.2 User Characteristics and Objectives

Assumed User Knowledge

- knowledge of how to download and install a desktop app.
- familiarity with rules of app(tag/subtag rule)
- Understanding of Spaced Repetition as a learning method

User Characteristics

- Students: Students of any discipline that may avail of spaced repetition. allowing a place to store and test yourself on keywords and topics.
- Project teams: with the dynamic tagging, it allows connecting many notes to an idea. As long as it's tagged correctly, every deviation into research can be linked back in.

User Objectives

- The ability to use spaced repetition which correctly cycles through notes.
- The ability to select which type of note to save as(Regular, Unfinished and Spaced repetition format)
- the ability to save and come back to any note
- the ability to store and search through any tag used
- user given the choice over which suggested note to open
- (desirable) The ability to share and edit the same note between users
- (desirable) choose templates for different hobbies e.g. gym plan

Describes the features of the user community, including their expected expertise with software systems and the application domain. Explain the objectives and requirements for the system from the user's perspective. It may include a "wish list" of desirable characteristics, along with more feasible solutions that are in line with the business objectives.

2.3 Operational Scenarios

1. **User creates account**
   Objective: A new user wants to sign up for Noted.
   Actions: User opens the app with no current login. They are asked to provide an email address or sign in.
   Successful End Condition: The user's account is successfully created and they are brought to the home screen.
   Unsuccessful End Condition: User provides invalid email address or attempts to add an already existing email.

2. **User creates new note**
   Objective: A user wants to create a note.
   Actions: User clicks create a new note and selects which type of note they want.
   Successful End Condition: The user creates a new note of their selected type
   Unsuccessful End Condition: The user has no room to create a new note and is prompted to delete any old notes.

3. **User saves a note**
   Objective: A user wants to save a note.
   Actions: User clicks save note and selects where they want to store it.
   Successful End Condition: The user saves their note in the selected location.
   Unsuccessful End Condition: The note is already saved and the user is prompted as such.

4. **User opens a note**
   Objective: A user wants to open a note.
   Actions: User clicks open note and selects which note they want to open.
   Successful End Condition: The user opens the note successfully.
   Unsuccessful End Condition: The user has no existing saved notes and is prompted as such.

5. **User clicks on an unfinished recommended note**
   Objective: A user wants to open a.past note they have not finished
   Actions: User clicks on one of the displayed unfinished notes to edit.
   Successful End Condition: The user successfully brings up the past note, being able to finish and mark as done.
   Unsuccessful End Condition: The user has not marked any past notes and the is prompted as such

6. **User views tags**
   Objective: A user wants to view existing tags saved.
   Actions: User clicks view tags and can scroll through existing tags
   Successful End Condition: The user can view subtags linked to the current tag being viewed.

Unsuccessful End Condition: The user has no saved tags and is prompted as such or the specific tag has no subtags linked.

7. **User dynamic views tags**
Objective: A user wants to dynamic view existing notes saved.
Actions: User clicks on note they want to view to and it displays on the right
Successful End Condition: The user can view note linked to tag
Unsuccessful End Condition: The user has no saved notes under that tag and is prompted as such.

8. **User dynamic views subtags**
Objective: A user wants to dynamic view existing notes saved.
Actions: User clicks on subtag they want to view and it displays note on right.
Successful End Condition: The user can view note linked to subtag
Unsuccessful End Condition: The user has no saved notes under that subtag and is prompted as such.

9. **User spaced repetition**
Objective: A user wants to see answer to spaced repetition question
Actions: User clicks on spaced repetition button to show answer
Successful End Condition: The answer is shown to user
Unsuccessful End Condition: no spaced repetition notes are written and user is informed

2.4 Constraints

1. **Time**: This project has a deadline due 7th May 2021 and as such the scope of this project must be achievable before this date. This can be achieved through delivering the projects features in stages, ensuring a working prototype will be available at the end of every sprint.

2. **Scope**: Due to the mentioned time frame, consideration must be made for the scope of this project. This will require consistent assessment to the development process of the apps base functionality and subsequent features will be added in order of importance.

3. **Performance**: The project must be able to process user requests in a quick manner, whether it is saving/opening a note, the dynamic note suggestion or the spaced repetition feature. This is a separate issue to User Experience or usability of the app, rather the reliability and speed of the app.

4. **User Experience** (UX):the app must be usable by any user with a base understanding of regular note taking apps like Google Docs. Each interactive elements function should be clear, specifically navigation buttons and the interaction with the dynamic suggestion.

3. Functional Requirements

Requirement	1
Description	The system must be able to create a user’s account
Criticality	Critical as all user activity is based off the assumption the user has an account
Technical issues	Implementation of user authentication and validation of all supplied user information
Dependencies	N/A

Requirement	2
Description	The system must be capable of storing user’s data
Criticality	Critical as all functionality involves storing and showing stored notes
Technical issues	Ensuring that all data is stored efficiently and logically on the database.
Dependencies	1

Requirement	3
Description	The system must be able to show spaced repetition notes over time
Criticality	Critical as spaced repetition is a core feature to the app
Technical issues	Must be able to keep track of

	time and order for each note
Dependencies	1,2

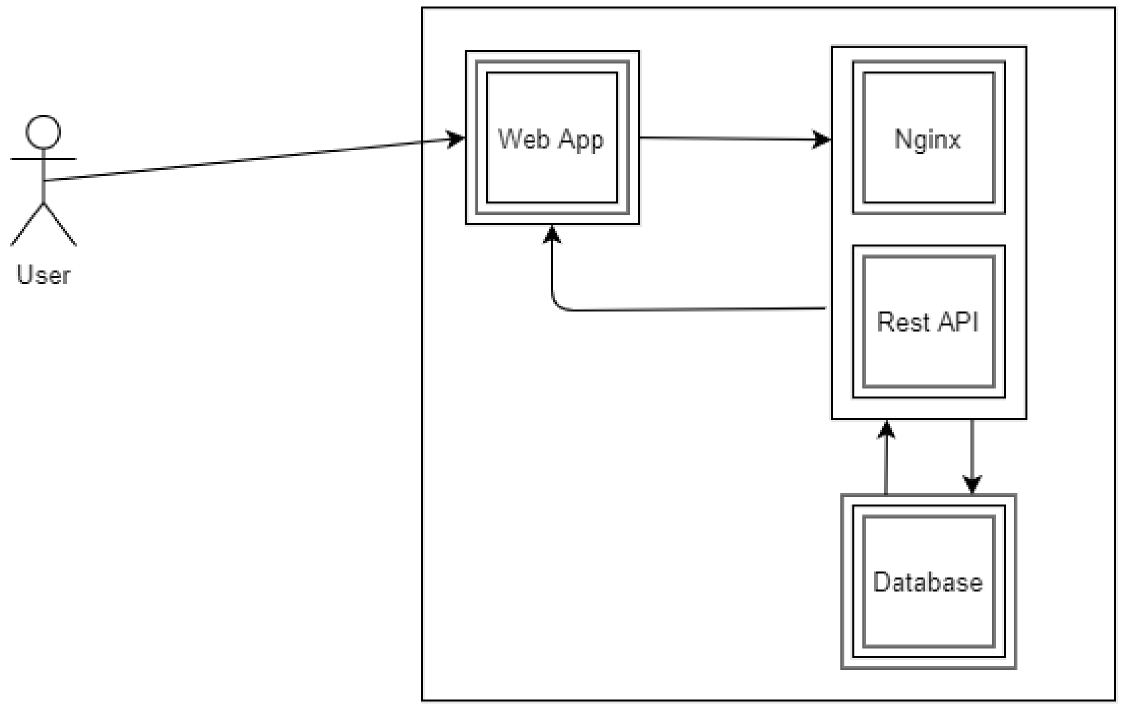
Requirement	4
Description	The system must be able to show past notes dynamically
Criticality	Critical as dynamic notes is a core feature to the app
Technical issues	Must be able to keep track of tags/subtags and properly display them for the user
Dependencies	1,2

Requirement	5
Description	The system must be able to display unfinished notes
Criticality	medium as it complements the app well but is not critical to its running
Technical issues	must be able to store and keep track of activity as to rank which ones to show first
Dependencies	1,2

Requirement	6
Description	The system must keep notes private from other users
Criticality	critical to the storing and security of users notes
Technical issues	must be able to store users notes separately from each other, removing access from each user
Dependencies	1,2

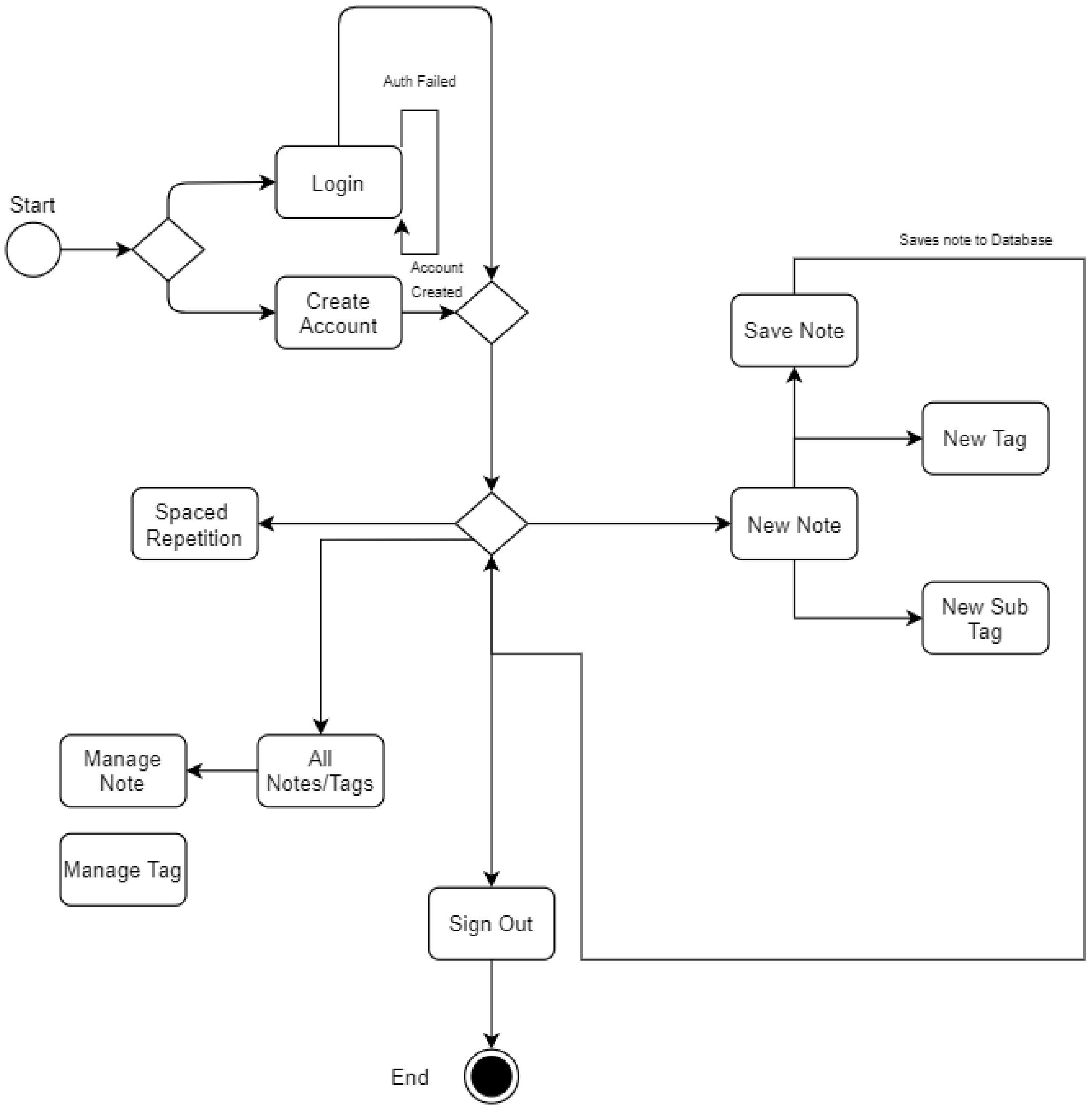
4. System Architecture

The user interacts with the web app front end, which in turn talks to the api to get/delete etc from the database.The boxes around each entity shows it has gone through containerisation and is within its own pod. this allows each element to act within one space but stay completely modular

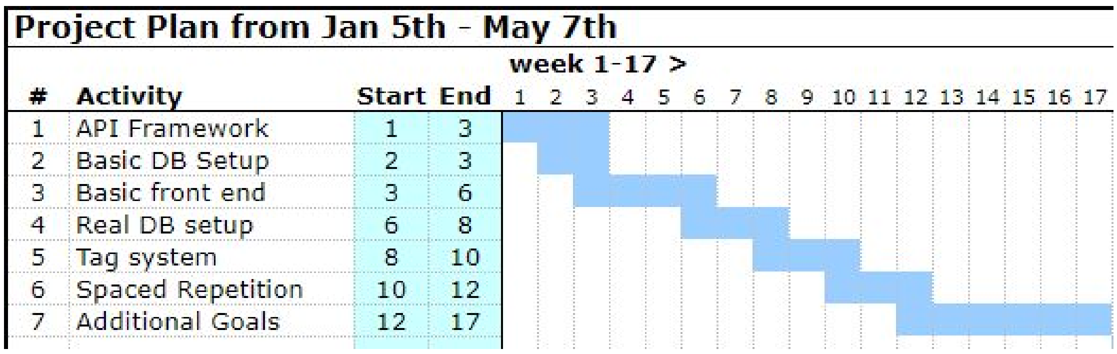


5. High-Level Design

Activity Model



6. Preliminary Schedule



Project Plan from Jan 5th - May 7th

				week 1-17 >																
#	Activity	Start	End	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	API Framework	1	3	■	■	■														
2	Basic DB Setup	2	3		■	■														
3	Basic front end	3	6			■	■	■	■											
4	Real DB setup	6	8						■	■	■									
5	Tag system	8	10								■	■	■							
6	Spaced Repetition	10	12										■	■	■					
7	Additional Goals	12	17												■	■	■	■	■	■

The goal is to have a minimum viable product by week 8, working on some of the actual functionality such as tag system. By week 12 i hope to have core functionality done leaving the last 5 weeks for stretch goals such as templates, multiple people working on one file etc…

7. Appendices

containerised definition:
https://searchitoperations.techtarget.com/definition/application-containerization-app-containerization

diagrams:
https://app.diagrams.net/