



School of Computer Science
The University of New South Wales
COMP1531 Software Engineering Fundamentals

UNSW Dreams

Software Requirements Report
20/04/2021

Team Blinker

David Liu - z5316672

Aleksander Kalcina – z5254925

Deepanshu – z5240759

Snigdha Anilkumar – z5270460

Saroj Sanjana Bombay – z5240805

Prepared for
COMP1531 Project development, UNSW
Iteration 3

Table of Contents

1. Introduction.....	3
2. Elicitation	3
2.1 Target user 1.....	3
2.2 Target user 2	4
2.3 Target user 3.....	4
3. Analysis & Specification – Use Cases	5
3.1 Target User 1.....	5
3.1.1 User Story.....	5
3.1.2 Use Case 1	6
3.2 Target User 2	6
3.2.1 User Story	6
3.2.2 Use Case 2	7
3.3 Target User 3	7
3.3.1 User Story	7
3.3.2 Use Case 3	7
4. Validation	8
4.1 Comment for User Case 1:.....	8
4.2 Comment for User Case 2:.....	8
4.3 Comment for User Case 3:.....	8
5. Interface Design	9
6. Conceptual Modelling (State)	13

1. Introduction

MS Dreams is a multifaceted application/website whose main objective is effective communication. This report outlines the capability and functionality of MS Dreams and the extent to which the stakeholders are satisfied with the application. UNSW/Team Blinker has come up with various features that not only exhibit the minimum functionality but helps in an effective and user-friendly means of communication.

Team Blinker has implemented the basic functionality features to run UNSW Dreams which helps the user to comprehend and enhance communication.

2. Elicitation

Interviewing Target Users who used/intended to use Dreams. We got the following responses for the survey questions. We interviewed three Users for elicitation.

2.1 Target user 1

Name – Shreya Anilkumar

Email – shreyasnigdha123@gmail.com

Q1) What tools do you frequently use for teamwork?

- MS Teams

Q2) What do you use this team-based app for?

- It's easy to collaborate and communicate with team members and it has a variety of features that help in building projects.

Q3) What problems did you find while using Dreams?

- In project management, assigning tasks and deadlines are important. So, lack of project management features is a downside.

Q4) What do you need from project-management feature?

- Taskbars to assign individual tasks and deadlines.

2.2 Target user 2

Name – Evaann Gupta

Email – evaannngupta.app@gmail.com

Q1) What tools do you frequently use for teamwork?

- MS Teams

Q2) What do you use this team-based app for?

- I used to use this app to contact my teammates for group courses. Now that I'm a recent graduate, I often use teams to talk my old professors or even the University if I have any questions about my graduation details for my further work

Q3) What problems did you find while using Dreams? Compared to MS Teams?

- Since MS Teams and Dreams are so closely comparable, I can point out a couple of things that Dreams lacks like sending attachments or videos as messages. One of the main problem that catches my attention is the option to be able to search for a specific user or channel.

Q4) Why do you want a user search feature?

- A lot of the times for course groups, it would become difficult to find a tutor or a specific person if I choose search for his page

2.3 Target user 3

Name – Neha Kutwal

Email – nehakutwal.3001@gmail.com

Q1) What tools do you frequently use for teamwork?

- MS Teams and Messenger

Q2) What do you use this team-based app for?

- As I'm an Economics student, I have a lot of courses that involve groups assignments. I use teams to access my course material and organize meetings with my group members. We often use Messenger to communicate as an alternative to teams.

Q3) What problems did you find while using Dreams? Compared to MS Teams or Messenger?

- As assignments have strict deadlines, sometimes we need to keep a check on the status of teammates like when they last came online or if they're online so that we can estimate if we'll be able to contact them etc.

Q4) Would you want to add any other additional feature?

- A calendar like teams where we can schedule meetings so that no one misses them out.

3. Analysis & Specification – Use Cases

3.1 Target User 1

3.1.1 User Story

- Shreya

As a user, I want to be able to create a taskbar that is accessible to all the members of the team so that we can assign tasks and deadlines.

User Acceptance Criteria:

- A New tab in the sidebar for tasks assigned.
- Tasks need to be able assigned in every channel.
- Users need to be able to access the taskbar.
- Users should be able to assign tasks and deadlines.
- Users can edit, remove and add tasks to the taskbar.

- Users should be able to see completed/removed tasks.
- Users should be able to see all related information about the task (e.g, deadlines in order of due-date, assignees, etc.)

3.1.2 Use Case 1

- **Use Case:** Assigning tasks to a user.
- **Goal in context:** Users need to be able to create and use taskbars for assigning tasks and deadlines. Also stores all the needed information.
- **Scope:** Channel
- **Level:** Subfunction
- **Preconditions:** User is logged in and a channel is created
- **Success End Condition:** A new tab in the sidebar of the channel and can store details and assign tasks to users.
- **Failed End Condition:** A taskbar cannot be accessed.
- **Primary Actor:** User
- **Trigger:** A tab in the sidebar of every channel named 'Taskbar'.

3.2 Target User 2

3.2.1 User Story

– Evaann

As a user, I want to be able to search for a specific user or a channel so it would be easier for me to know if the person is on the platform or on the same channel as me.

User Acceptance Criteria :

- A search bar that allows searches for Users
- A search bar that allows searches for Channels
- Users should be able to see if they have any channels in common
- A dm option should be available on the user's page to send a quick dm

3.2.2 Use Case 2

- **Use Case:** Search for a specific user or channel
- **Goal in context:** Users need to be able to search for other users on the search bar to find them.
- **Scope:** Channel and Auth
- **Level:** Subfunction
- **Preconditions:** User is logged in and exists in channel. User is in a common channel or dm as the user they are searching for.
- **Success End Condition:** User able to view the result of the search. Details of the user or takes them to channel.
- **Failed End Condition:** Cannot search on the search bar
- **Primary Actor:** User
- **Trigger:** A search bar at the top of the home page.

3.3 Target User 3

3.3.1 User Story

– Neha

As a student, I want to be able to view the activity status of my group members so that I know if I could get in touch with them soon.

User Acceptance Criteria :

- Activity statuses in different stages
- Shows when user is online
- Shows when user is offline
- Shows when user has been online till 12 hours ago
- Stages are color-coded
- Shows up beside the users' names

3.3.2 Use Case 3

- **Use Case:** Show activity status of the user
- **Goal in context:** Users need to be able to see the login status of users' they are in mutual with
- **Scope:** Auth
- **Level:** Subfunction
- **Preconditions:** User is logged in and in a common channel or dm as the user they can see.
- **Success End Condition:** User able to view the activity status of another user
- **Failed End Condition:** Activity status not shows
- **Primary Actor:** User
- **Trigger:** A colour code with the User's name

4. Validation

4.1 Comment for User Case 1:

Adding the taskbar for each channel solved the issue of not having a to-do list in the app. Thus seeing the task bar of not just the current tasks but the previous ones helped in improving organising and understanding where each user stands. This solves my main issue.

4.2 Comment for User Case 2:

I think the use case sums up the problem. With the solution, I will be able to individually search for a person rather than go to specific channels or dms. It would save me a lot of time

4.3 Comment for User Case 3:

The use case describes my problem well as it's also a quite simple problem. Being able to see the activity status of my group members will help us to keep a track of each other and this function will do just that.

5. Interface Design

Data Types:

task_id (int) – uniquely generated id for each task created

items ({ }) – is a dictionary which holds the details of the specified task
 { task : task_id, assignees[], due_date, is_complete }

is_complete (bool) – is a Boolean which tells us if the task is completed or not

assignees ([]) – is a list of users who's been assigned specific tasks

due_date (str) – is a string that tells us the deadline

status (bool) – Is a Boolean which tells us if a user is online or offline

time_out (int) – Keeps track of the time after logout

edit (str) – A description of the edited task

NOTE : Taskbar is the list of all tasks and each individual task has details called items

Table 1 : Interface Design for all functions

Name & Description	HTTP Method	Data Types	Exceptions
taskbar/create Creates a new taskbar list for a given channel that returns a unique id for the taskbar	POST	Parameters: {token, channel_id} Return Type: {task_id}	InputError: <ul style="list-style-type: none"> Taskbar list already exists for the channel AccessError: <ul style="list-style-type: none"> When user is not a member of the channel

<p>taskbar/list</p> <p>Provides a list of incomplete tasks in a specified list along with their associated details</p>	GET	<p>Parameters: {token, task_id}</p> <p>Return Type: {task_id, items}</p>	<p>InputError: Invalid task_id</p> <p>AccessError: When user is not a member of the channel</p>
<p>taskbar/listall</p> <p>Provides a list of all the tasks (complete and incomplete) along with their associated details.</p>	GET	<p>Parameters: {token, task_id, is_complete, assignees, due_date}</p> <p>Return Type: {task_id, items}</p>	<p>InputError: <ul style="list-style-type: none"> Invalid task_id </p> <p>AccessError: <ul style="list-style-type: none"> When user is not a member of the channel </p>
<p>item/add</p> <p>Creates a new task list for a given channel that returns a unique id for the task.</p>	POST	<p>Parameters: {token, task_id, due_date, assignees }</p> <p>Return Type: {item_id}</p>	<p>InputError: <ul style="list-style-type: none"> Taskbar list already exists for the channel </p> <p>AccessError: <ul style="list-style-type: none"> When user is not a member of the channel </p>

<p>item/details</p> <p>Given item_id, provide basic details about the task.</p>	GET	<p>Parameters: {token,item_id}</p> <p>Return Type: {status, assignees, due_date}</p>	<p>InputError:</p> <ul style="list-style-type: none"> Item_id does not refer to existing task <p>AccessError:</p> <ul style="list-style-type: none"> When user is not a member of the channel
<p>item/update</p> <p>Given item_id, update the status of the task with is_complete boolean</p>	PUT	<p>Parameters: {token,item_id, is_complete}</p> <p>Return Type: {}</p>	<p>InputError:</p> <ul style="list-style-type: none"> Item_id does not refer to existing task <p>AccessError:</p> <ul style="list-style-type: none"> When user is not a member of the channel
<p>Item/edit</p> <p>Given item_id update task description.</p>	PUT	<p>Parameters: {token,item_id, edit}</p> <p>Return Type: {}</p>	<p>InputError:</p> <ul style="list-style-type: none"> Item_id does not refer to existing task <p>AccessError:</p> <ul style="list-style-type: none"> When user is not a member of the channel

<p>Search/user</p> <p>Given a user handle or channel name, get details of the user or return the channel</p>	GET	<p>Parameters: {user_handle, name}</p> <p>Return Type: {user_id, channel_list}</p>	<p>InputError:</p> <ul style="list-style-type: none"> Handle does not refer to existing user <p>AccessError:</p> <ul style="list-style-type: none"> When user is not registered
<p>Search/send_dm</p> <p>Given a user handle, find the user first and send a dm directly</p>	PUT	<p>Parameters: {user_handle, auth_user_id, message}</p> <p>Return Type: {dm_id, message_id}</p>	<p>InputError:</p> <ul style="list-style-type: none"> Handle does not refer to existing user <p>AccessError:</p> <ul style="list-style-type: none"> When user is not registered When dm already exists
<p>Auth/online</p> <p>Given a user's last token, checks whether the token is still in session</p>	GET	<p>Parameters: {token, user_id}</p> <p>Return Type: {status}</p>	<p>InputError:</p> <ul style="list-style-type: none"> User_id does not refer to existing user <p>AccessError:</p> <ul style="list-style-type: none"> When user is not registered

<p><u>Auth/last_seen</u></p> <p>Given that user's token turns invalid (log out), starts the time and tracks up to 12 hours</p>	PUT	<p>Parameters: {token,user_id}</p> <p>Return Type: {time_out}</p>	<p>InputError:</p> <ul style="list-style-type: none"> User_id does not refer to existing user <p>AccessError:</p> <ul style="list-style-type: none"> When user token is still valid
--	-----	---	---

6. Conceptual Modelling (State)

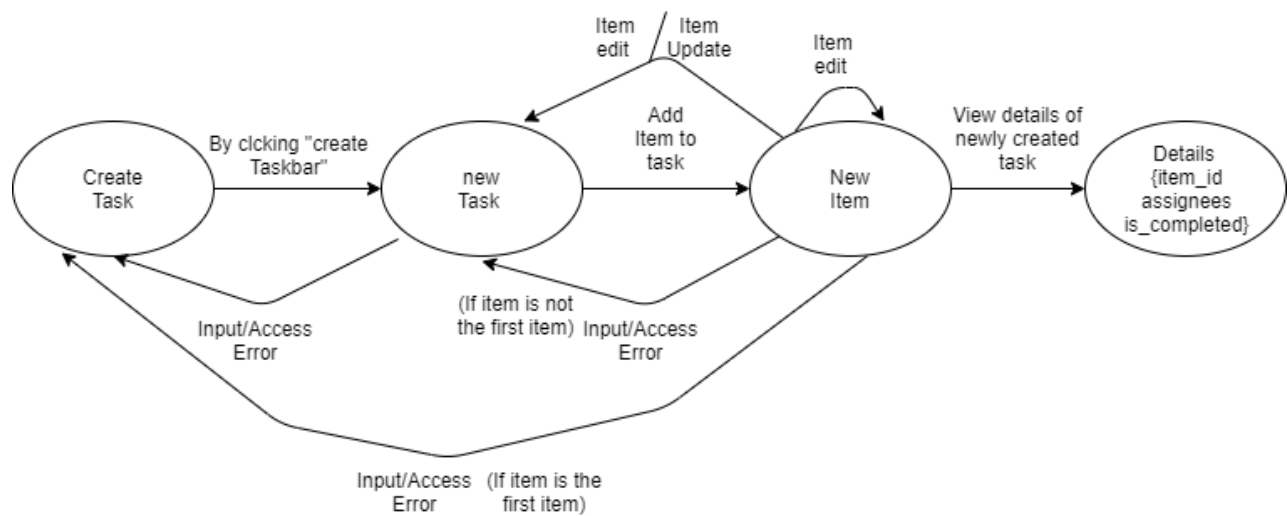


Figure 1: State diagram for creating taskbar

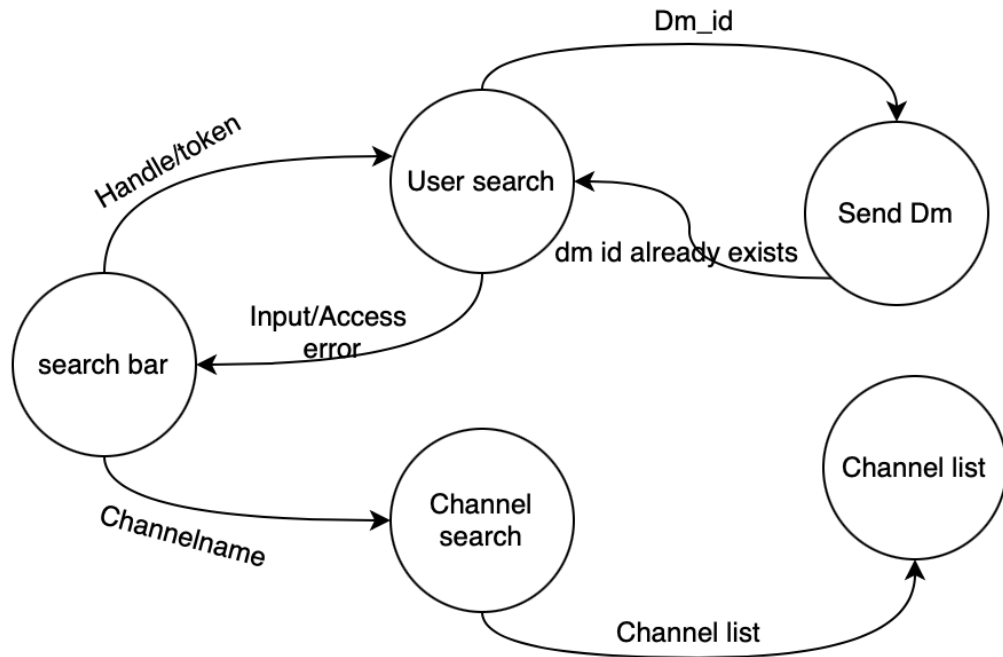


Figure 2 : State diagram for search

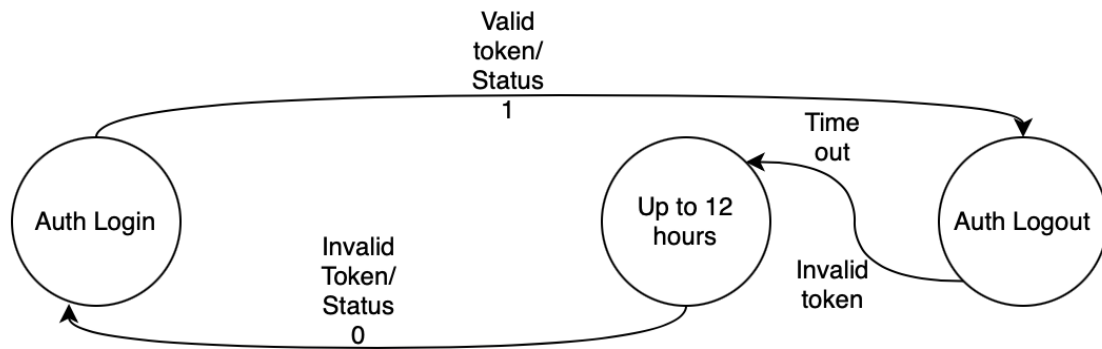


Figure 3 : State diagram for login status