

**Study Buddy**

21-Aug-2022

Aung Ko Ko Khant (A0249480X)

Aye Mya Phoo (A0249473U)

Cheong Li Yuan (A0144049H)

Khan Sher Mostafa Somik (A0249468L)

Nyan Htet Win Maung (A0249498E)

Xie Ting Ting(A0249501J)

Zhao Ziyou (A1084806X)

Contents

[Product Introduction 4](#_Toc111919519)

[Background 4](#_Toc111919520)

[Features Developed 5](#_Toc111919521)

[Android 5](#_Toc111919522)

[General 5](#_Toc111919523)

[Study Groups / Chats 5](#_Toc111919524)

[Events 5](#_Toc111919525)

[Books 5](#_Toc111919526)

[Profile 5](#_Toc111919527)

[React 6](#_Toc111919528)

[General 6](#_Toc111919529)

[Study Groups / Chats 6](#_Toc111919530)

[Events 6](#_Toc111919531)

[Books 6](#_Toc111919532)

[Profile 6](#_Toc111919533)

[ML(Python) 6](#_Toc111919534)

[Sentiment Analysis using BERT model 6](#_Toc111919535)

[Book recommendation order using: 6](#_Toc111919536)

[Book search top 5 results using: 7](#_Toc111919537)

[Java Backend API 7](#_Toc111919538)

[Authentication (JWT Token) 7](#_Toc111919539)

[User 7](#_Toc111919540)

[Message 7](#_Toc111919541)

[Group (PUBLIC, PRIVATE) 7](#_Toc111919542)

[Event 7](#_Toc111919543)

[Book 8](#_Toc111919544)

[Deployment on Oracle Cloud 8](#_Toc111919545)

[Improvements Planned 8](#_Toc111919546)

[Technical Challenges/Lesson Learnt 9](#_Toc111919547)

[Android 9](#_Toc111919548)

[Utility Classes 9](#_Toc111919549)

[Widget Class 9](#_Toc111919550)

[Adapter Classes 9](#_Toc111919551)

[Services 9](#_Toc111919552)

[React 9](#_Toc111919553)

[React Hook 9](#_Toc111919554)

[Client Site Routing 9](#_Toc111919555)

[CORS issue 9](#_Toc111919556)

[Styling issue 10](#_Toc111919557)

[HTTPS issue 10](#_Toc111919558)

[Java Backend API 10](#_Toc111919559)

[Authentication 10](#_Toc111919560)

[General 10](#_Toc111919561)

[ML(Python) 10](#_Toc111919562)

[Deployment on Oracle Cloud 10](#_Toc111919563)

[Bibliography 11](#_Toc111919564)

# Product Introduction

The StudyBuddy project consists of an Android app and React app for users. Technologies used in the project includes Android (Java), React (ReactJS), Spring Boot (Java), Python (Flask) and Docker containers.

StudyBuddy was created to serve any user who struggle with learning by providing them a platform for discussion. Users can create and join events to initiate discussions on difficult topics. Machine learning algorithms work in the background to recommend users with textbook resources according to their interests and recently joined chat groups.

StudyBuddy works seamlessly on both its Android app as well as React app, giving users options to choose as they prefer.

# Background

Everyone understands the struggles of learning. To be stuck on a topic with unfathomable answers from Google and no one else to ask can be particularly daunting. Forums replies might take days to reply, which adds to the isolation of such platforms.

StudyBuddy aims to bring such discussions to a live platform where students can share their questions, doubts, and answers with one another to facilitate learning and make it a more enjoyable journey.

Our app is catering to:

* Users who are looking for a platform to discuss difficult topics they are facing.
* Users who are looking to schedule events with other users to facilitate chat discussion
* Users who are looking for book recommendations based on their interests or current difficult topics

# Features Developed

## Android

### General

1. Register for new account.
2. Login to app.

### Study Groups / Chats

1. See all enrolled study groups.
2. Tap on study group to view/send messages in the group.
3. New message notifications. Either tap on notification to open the group to view the message or navigate to the group manually.
4. Able to send/receive multimedia messages containing images, videos, or attachments files.
5. Able to view images/play videos in full screen by tapping on them from the chat screen.
6. Able to download files sent.
7. Supports Gboard (Google Keyboard) and is able send reaction gif images or stickers built into the Keyboard by choosing them from the keyboard.
8. Long pressing on any chat message will copy its contents to the clipboard.
9. Search for users to chat one to one.
10. Tapping on the user will open a private chat group with the selected user.
11. Tapping on own name in the list will also open their profile page.

### Events

1. View all available events.
2. Join events. Joining event will also enrol the user in the study group.
3. Create new events to host.

### Books

1. View books recommended by ML based on selected interests & groups joined.
2. Search for books to read.
3. Bring to the books page by tapping on the book.

### Profile

1. View user profile.
2. Upload new profile photo by tapping on the profile image.
3. Edit interests.
4. Change password.
5. Logout.

## React

### General

1. Register for new account
2. Login to app

### Study Groups / Chats

1. Study groups with chatting.
2. Able to send images/videos/files.
3. Able to view images/play videos in full screen by tapping on them from the chat screen.
4. Able to download files sent
5. Search for users to chat 1 to 1
6. Both light mode and dark mode available
7. Mobile responsive for small device
8. Auto scroll to last message when click on each chat list.
9. Display message send time based on current timestamp.
10. Emoji supported
11. Hover on the user photo will pop up user profile and tapping on user profile will open private chat with user.

### Events

1. View all available events with Calendar View
2. Join events or View Discussion (already member of group) by clicking each event
3. Create new events based on Date and Time
4. Have Month View, Week View, Day View and Agenda View.

### Books

1. View books recommended based on interests & groups joined
2. Search for books with autocomplete feature
3. View books

### Profile

1. View current profile
2. Change profile photo by clicking upload photo (Drag and Drop supported)
3. Edit interests, email, display name
4. Change password
5. Logout

## ML(Python)

### Sentiment Analysis using BERT model

### Book recommendation order using:

1. Picked interests
2. Cosine similarity of interests and latest event joined against textbook descriptions
3. Supported by scraped textbook data from <https://open.bccampus.ca/browse-our-collection/find-open-textbooks/>

### Book search top 5 results using:

1. Cosine Similarity against textbook descriptions and title

## Java Backend API

Before using the resource API endpoints, please make sure your account is registered in the server as one of the users. If not register yet, you can use "Register user " endpoint to register.

### Authentication (JWT Token)

1. JWT is used for communications between clients and server. The client needs to have correct username and password when getting token from server. When the username and password is correct, server will give token for authentication purposes. The client needs to use the JWT token every time when communicating with resources API. The JWT token expiration time is 90 days.
2. You can get the JWT token from folder name "JWT generate token" and use "generate token".

### User

1. To register the new user, use "register user".
2. To update the user information, use "update user".
3. To update the password, use "update password".
4. To edit user interest, use "edit user interest".

### Message

1. To send the message for both public and private group, use "create message in group".
2. To retrieve messages in both public and private group with limit and offset, use "retrieve messages by groupId with limit and offset". OR to retrieve all messages in both public and private group, use "find Messages by groupId".

### Group (PUBLIC, PRIVATE)

1. To create PUBLIC Group, use " create new group".
2. To create PRIVATE Group, use "create new or retrieve existing individual group with two users". (NOTE: It will give existing group when private with two participants exist)
3. To search the group which user join as participant, use "search group by single user as participant".

### Event

1. To create Event, use "Create Event". (NOTE: It will also create public group)
2. To join the event as user, use "user join event".
3. To search the specific event, use "search event by Id".
4. To get all existing events, use "get all events".

### Book

1. To search the book, use "find book by Id".
2. To add new book, use "create book".
3. To search the book by "title", use " search book with title".
4. To search the book ordered by use, use “get user ordered book”

## Deployment on Oracle Cloud

1. Easy deployment to cloud with 3 easy commands (given “git” and “docker” pre-installed)

git pull

docker compose build

docker compose up -d

1. Created custom Docker files for building
   1. Java Spring Boot application
   2. React JS on node.js
   3. Python on flask
2. Created docker-compose.yml file to stack the docker containers along with MySQL and build bridged networks between
   1. Java Spring Boot and MySQL server
   2. Java Spring Boot and Python ML server
   3. phpMyAdmin DB manager to MySQL server

# Improvements Planned

1. Ability to edit and delete messages
2. Filter and hide swear words
3. Dark mode on Android
4. Improve dashboard UI on React
5. Able to kick out user from group

# Technical Challenges/Lesson Learnt

## Android

### Utility Classes

* Custom APIRequestHandler class to talk to API using raw HTTP commands. The method postFile was specially challenging as a multipart post request was needed to be created for this purpose.
* Custom APICommunication class to wrap custom class APIRequestHandler, run it in a background thread, retrieve results from it and parse JSON replies.

### Widget Class

* Custom GifEditText to override android EditText to integrate with GBoard (Google Keyboard) gif/sticker support.

### Adapter Classes

* Custom ArrayAdapter class to support listview to:
  1. display books in BookListAdapter
  2. display groups in GroupListAdapter
  3. display users in UserListAdapter
* Custom RecyclerView Adapter class to
  1. display events in EventListAdapter with 3 different formats
  2. display chats in MessageListAdapter with 6 different formats

### Services

* Custom Foreground service to poll groups for new messages

## React

### React Hook

* React hook is a bit tricky. If API request is not handled properly using hook effect, there will be memory heat problems.
* As React hook is newly introduced feature, it takes a bit of time to understand and use it.

### Client Site Routing

* react-router-dom V6 is used for client site routing. It takes a bit of time to know how to use the library.

### CORS issue

* This issue always happens when server and client site are not in the same domain. Server will reject every request from client because of CORS issue.
* Issue was overcome by setting up Cors's compatibility setting in backend code.

### Styling issue

* To support multi-mode in chat style, style was created using sass instead of CSS. Usage of sass eliminated a lot of duplicate code.

### HTTPS issue

* To send data over the internet, browser complains unless it is being sent over a HTTPS tunnel. So, react is running behind a Nginx reverse proxy providing HTTPS support.
* If react is running with HTTPS, it can't send data to JAVA backend as it was running without HTTPS. So now all the servers are behind Nginx reverse proxies.

### Auto Scroll

* Auto scroll chat history when there is new message especially image in React. When message is arrived, auto scroll will be triggered but image is still loading, and browser don't know exact height to be scrolled yet.
* So able to solve issue by adding auto scroll function inside onload event of image.

### Sate Management

* React UI state management. To maintain single source of truth between Parent and Child components.

## Java Backend API

### Authentication

* Effort was taken to ensure all the server communication pass through the " JWT Authentication Entry Point”.
* An authentication error message will be shown when JWT token is expired.

### General

* Circular references in entity was creating problems and were removed in the middle of development.
* Avoiding null exceptions in getting Object with proper checks.
* Meeting datelines and required Business logic.

## ML(Python)

* Scraping of textbooks is done programmatically but selection of data is done manually. Any changes to website format might break the written web scraping code, which might prevent reproductivity or reusability.
* As the code for cosine similarity is adapted from the workshop, the challenge is to modify it to work on the textbooks data and serve the function as a API call.

## Deployment on Oracle Cloud

* Creating a linux server on the cloud, accessing it through SSH without using passwords and installing docker with compose plugin.
* Opening required ports on the server.
* Managing the server through ssh commands only
* Building the containers on the server and connecting them without opening publicly accessible ports to database or ML.
* Running separate Nginx server on docker to provide HTTPS support to backend API and React frontend pages.

# Bibliography

*99designs*. (n.d.). Retrieved from https://99designs.com.sg/logo-design/contests/study-buddy-has-sloth-logo-promote-fun-side-509210