NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

System Requirement Specifications (SRS)

Kim’s Convenience Modern Cinema

Wong Ying

Hussain Khozema Kheriwala

Desmond Yap

Duc Ta

Jeremy Book

Taneja Parthsarthi

Team Kim’s Convenience

School of Computer Science and Engineering, Nanyang Technological University

Revision History

Author(s)	Date	Description of Changes	Version #
Wong Ying Desmond Yap Duc Ta	03 September 2021	Preliminary Version	1.0
Wong Ying Hussain Khozema Kheriwala Desmond Yap Duc Ta Jeremy Book Taneja Parthsarthi	07 September 2021	Preliminary discussion and review of entire document	1.1
Wong Ying Desmond Yap Duc Ta	10 September 2021	Update discussed areas	2.0
Wong Ying Hussain Khozema Kheriwala Desmond Yap Duc Ta Jeremy Book Taneja Parthsarthi	14 September 2021	Final review	2.1

Table of Contents

1. Problem Statement

With the introduction of many video streaming services in Singapore, many traditional movie cinemas are struggling to compete and retain their market share. These global video streaming services offer viewers with extensive libraries of on-demand movies and TV shows to watch. However, these services greatly rely on their recommender system to retain their viewers. As such there is a need for cinema chains in Singapore to employ such a recommender system as well in order to retain their market share.

2. Overview

2.1 Background

Seeing as how visiting the cinemas is one of Singaporeans favourite pastime, it is not surprising to see that many global video streaming services such as Disney Plus and Netflix, have since made their way into our shores. Last year alone, Singapore's box office generated S$41.3 million in revenue [1]. Coupled with the fact that many cinemas faced stricter regulation for movie viewing due to the ongoing Covid-19 pandemic, the video streaming landscape in Singapore is only going to experience further expansion. With such a wide selection of video streaming service providers, many individuals would rather choose to watch movies or TV shows on these platforms in the comforts of their homes instead of a traditional movie viewing experience in a cinema. This is also evident from the fact that there is a significant decrease in the total number of visits to cinemas in 2020 than in 2019 [2]. One of the main distinguishing features that sets apart streaming platforms from traditional cinemas is the existence of a recommender system. One study showed that 80% of stream time is achieved through Netflix’s recommender system [3]. Currently, all cinema chains in Singapore do not provide such a system. A recommender system employs machine learning on user generated data to provide a list of movie recommendations for the user. This user data can be generated from user activities such as past movies watched or their bookmarked movies. As such KCMC aims to help local cinema businesses bridge this gap in order to remain competitive and to recover from the impact of the pandemic by introducing a recommender system.

2.2 Overall Description

Kim's Convenience Modern Cinema (KCMC) is a movie booking web application which provides personalised movie recommendations with the help of deep learning to create a recommendation system. The web application provides a clean and sleek user interface for users to navigate and book movies and is developed using Python's Django web development framework. Other features include user authentication and personalised movie recommendations based on previous watch history. SQLite will be the main database used for storing movies, shows, booking and user data. Most importantly the web application will consist of a recommendation system model trained with cinema goers' booking history data. The model will be trained on a nightly basis.

3. Investigation & Analysis Methodology

3.1 System Investigation

The web application will capture the inputs from cinema administrators and process it before passing it on to the backend server. The server request is first authenticated to ensure that the request is generated from a secure source and after it will be updated in the SQLite database at an appropriate database schema. Once the request has successfully completed, a transaction indicator will be sent back to the web application to indicate whether the transaction was successful.

On the other hand, when users wish to browse movie selections, halls or shows, an appropriate query request is sent from the web application to the system. Similarly, the request is first validated and authenticated before the query is sent to the database. If the query is successful, , the results of the user's query will be sent back to the web application and displayed for the user to see.

3.2 Analysis Methodology

3.2.1 Feasibility study and requirements elicitation

- Organise a development team composed of experts in back-end development and web application development
- A series of interviews with personnel and IT staff working directly with cinema chain websites to gather feedback and to define the current technological environment and future system needs.
- Gather feedback from frequent cinema goers to better understand the their needs in order to appropriately design the user interface
- Gather feedback from avid Netflix, Disney+, AmazonPrime users to gain insight into why they choose to use these services. These insights could be used to incorporate design changes into cinema websites to increase their viewership.
- A Feasibility and Risk Assessment study will be carried out to establish which solution(s) are the most viable and applicable based upon the outcomes of the interviews.

3.2.2 System analysis and requirements specification

3.2.2.1 Perform an analysis of the problem using object-oriented techniques

The Unified Modelling Language (UML) will be used to represent an external view of the application model. It consists of movie records, hall information, user details and user viewing data. This System requirement Specifications document constitutes a portion of the documentation for the project. Several desired features of the new system are listed below.

User:

- Account creation and authentication;
- Viewing movies and showtimes;
- Booking shows at selected showtime;
- Displaying recommended movies for the users.

Cinema Administrators

- Add or remove movies and showtimes
- Making adjustments to the recommender system

3.2.2.2 Scope and Limitations

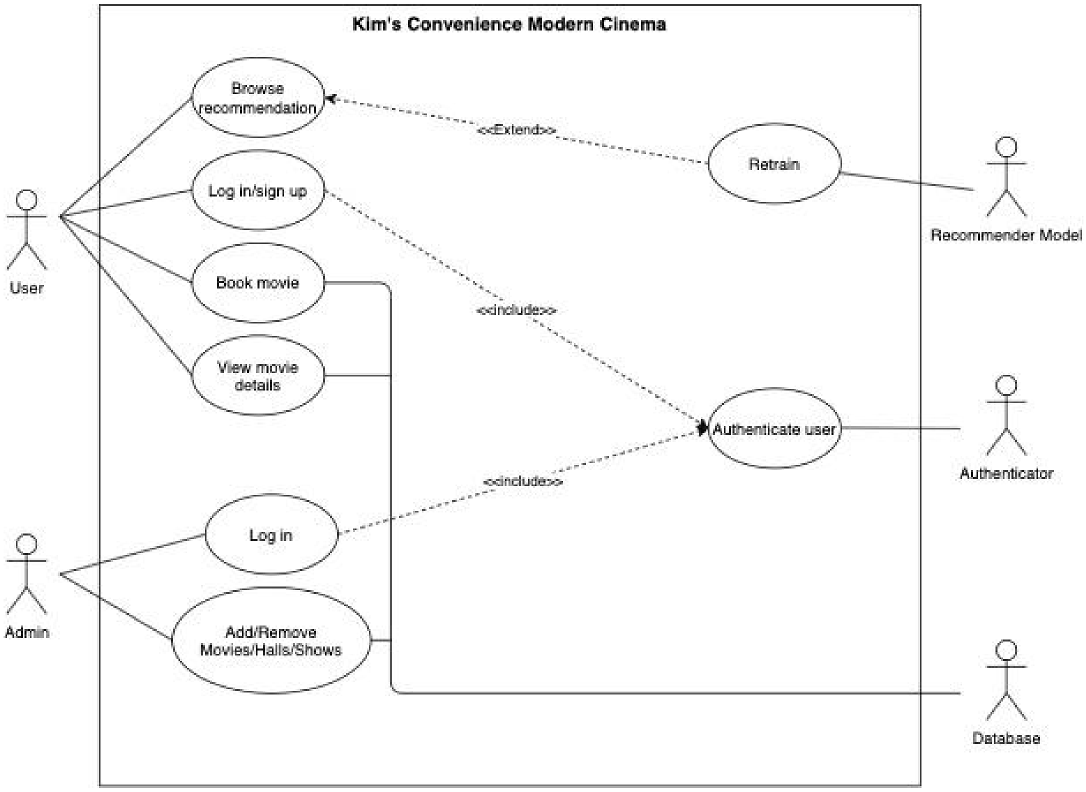
Analysis methodology will involve several components, namely, business analysis, requirement analysis, data analysis, process analysis (web), and application architecture:

- **Business analysis**: Information pertaining to the business rule, business system interface, function, ownership, sponsorship and project budget requirement can be found in the project proposal. Please refer to the project proposal for more information.
- **Requirement analysis**: Analysing the system requirements, user requirements definition for web interface and functional and nonfunctional requirements.
- **Data analysis**: Involves process of collection of movie goers data, data definition, validation, storage, manipulation and retrieval in the database. The most crucial data would be the user data as it is the foundation of the recommender system.
- **Process analysis**: Study, analyze and decompose the data and process flow when users interact with the web application in order to design an intuitive user interface and manage the data transfer processes.
- **Application architecture**: Analyze the application data structure, useability, feasibility, interface design and interaction and its implementation.

3.2.3 Object-oriented design using UML

A detailed and comprehensive object-oriented design for the movie booking platform will be developed. UML will be used again for the graphical representation and documentation of the design. The platform's primary responsibility is to allow users to view recommended movies and make movie bookings. In essence, the platform will display the current list of movie screenings and their recommended movie list and allow users to select a movie and their preferred show timing. Additionally, cinema administrators will be able to add movies, halls or shows by filling up a web based form that will be processed by the SQLite back-end system. The system will be secured with user's identification and password.

3.2.3.1 Use Case Diagram for Web Application



3.2.4 Prototyping

The team will be using Object Oriented Evolutionary Prototyping to implement a limited function prototype of the movie booking platform. The prototype is aimed to be a fully functional demonstration of the movie booking platform. However, as the prototype is only used as a proof of concept purpose, it will not contain all the features that have been planned. The prototype presented is by no means the final product delivered. The prototype will be presented to the implementation team for further testing and refinement. Below is a list of functions that the prototype will be expected to have.

1. User account creation and login
2. Movie recommendation system
3. Movie and Show browsing
4. Movie booking
5. Editing of Movie and Show details

4. Constraints

4.1 Proprietary Hardware

The movie booking web application only requires a remote backend server such as SQLite in order to store the necessary data for the application to run.

4.2 Scalability

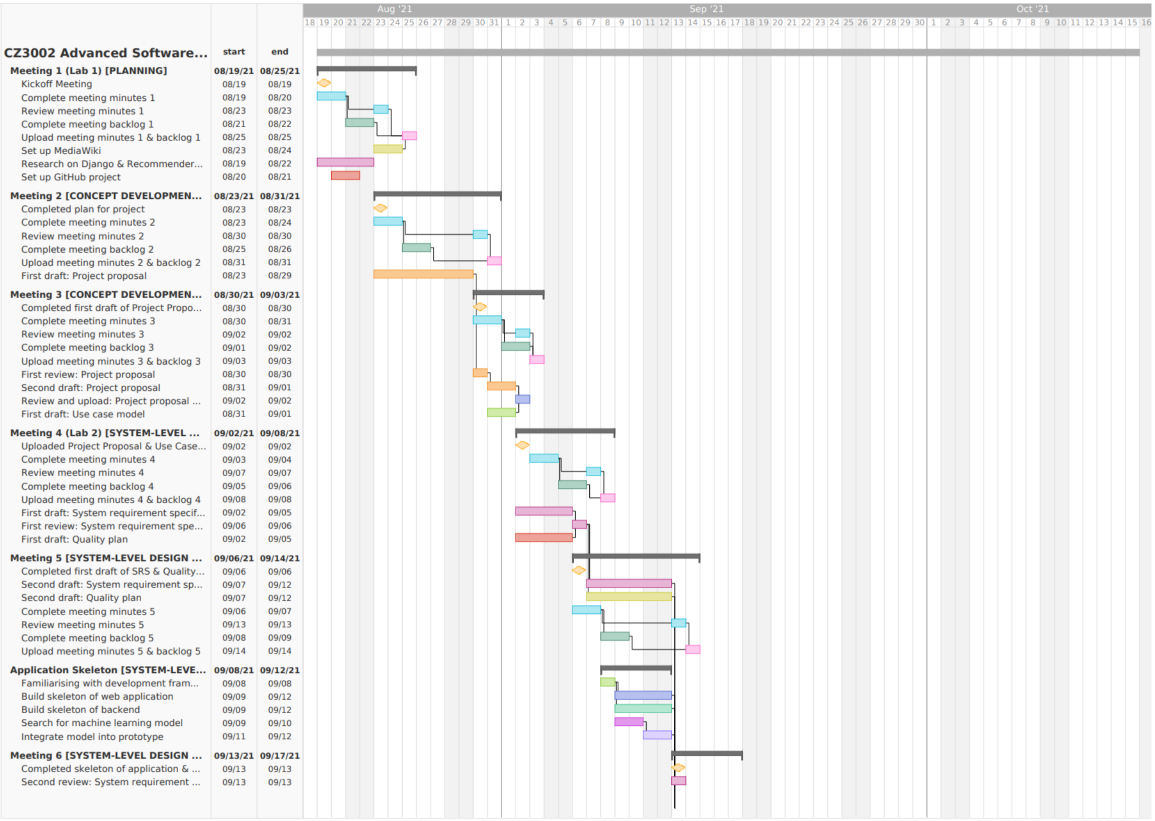
As the application is dependent on the backend server to retrieve the necessary information, the scalability of the application will depend on the backend server used. If the backend server is able to scale up to accommodate multiple users, the application will too as well.

4.3 Retrain Of Model

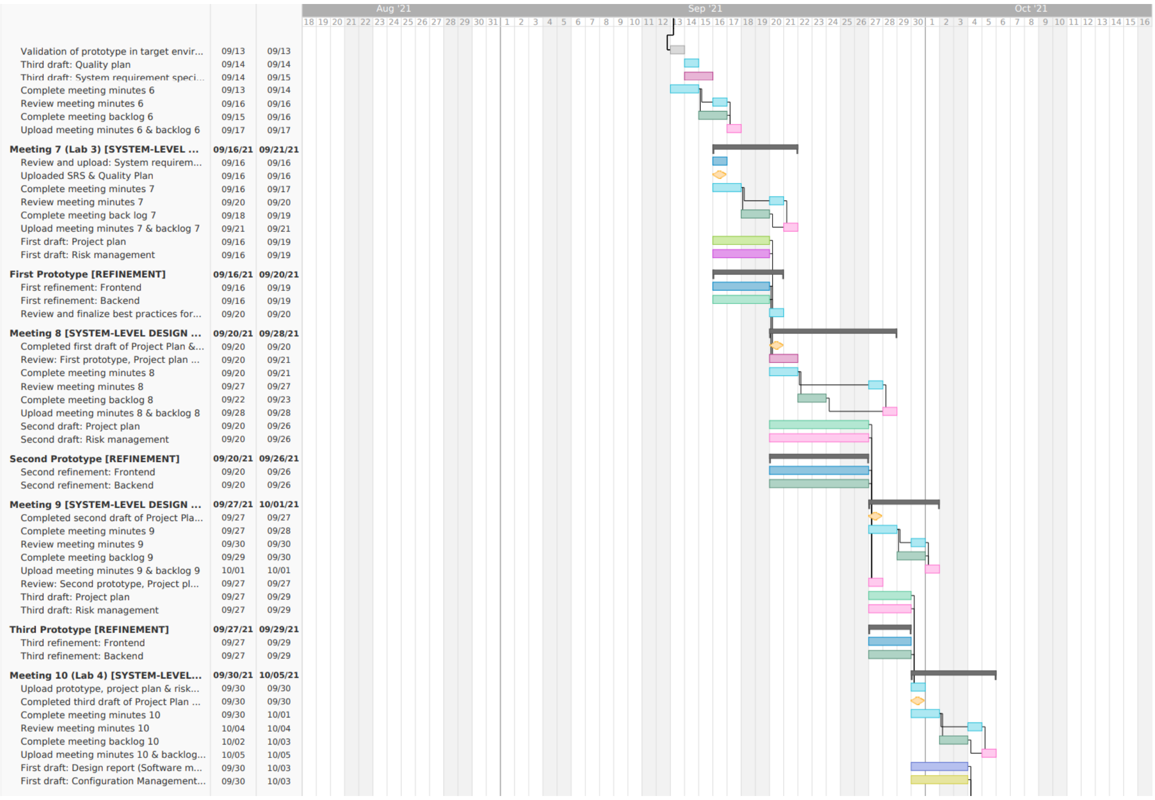
New registered users must at least have one movie booking as an input to the recommender model system.

4.4 Project Schedule

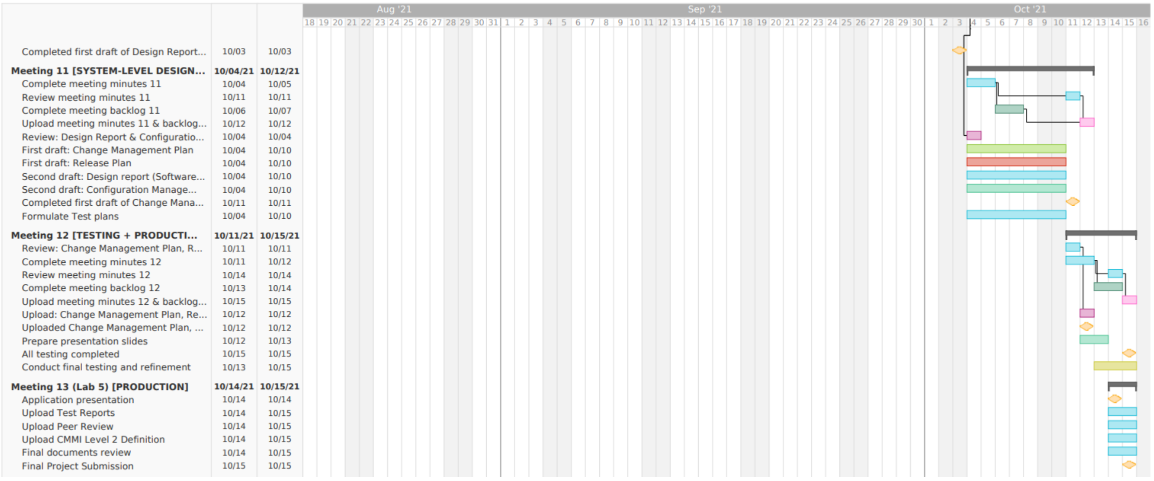
The project has an estimated timeframe of two months. The project is broken into seven task phases, namely, planning, concept development, system-level design, detailed design, testing and refinement and production. In the final phase, the production phase, the final prototype will be demonstrated and presented. Detailed description of each phase can be found in the project proposal.



CZ3002 Advanced Software...	start	end
Meeting 1 (Lab 1) [PLANNING]	**08/19/21**	**08/25/21**
Kickoff Meeting	08/19	08/19
Complete meeting minutes 1	08/19	08/20
Review meeting minutes 1	08/23	08/23
Complete meeting backlog 1	08/21	08/22
Upload meeting minutes 1 & backlog 1	08/25	08/25
Set up MediaWiki	08/23	08/24
Research on Django & Recommender...	08/19	08/22
Set up GitHub project	08/20	08/21
Meeting 2 [CONCEPT DEVELOPMEN...	**08/23/21**	**08/31/21**
Completed plan for project	08/23	08/23
Complete meeting minutes 2	08/23	08/24
Review meeting minutes 2	08/30	08/30
Complete meeting backlog 2	08/25	08/26
Upload meeting minutes 2 & backlog 2	08/31	08/31
First draft: Project proposal	08/23	08/29
Meeting 3 [CONCEPT DEVELOPMEN...	**08/30/21**	**09/03/21**
Completed first draft of Project Propo...	08/30	08/30
Complete meeting minutes 3	08/30	08/31
Review meeting minutes 3	09/02	09/02
Complete meeting backlog 3	09/01	09/02
Upload meeting minutes 3 & backlog 3	09/03	09/03
First review: Project proposal	08/30	08/30
Second draft: Project proposal	08/31	09/01
Review and upload: Project proposal ...	09/02	09/02
First draft: Use case model	08/31	09/01
Meeting 4 (Lab 2) [SYSTEM-LEVEL ...	**09/02/21**	**09/08/21**
Uploaded Project Proposal & Use Case...	09/02	09/02
Complete meeting minutes 4	09/03	09/04
Review meeting minutes 4	09/07	09/07
Complete meeting backlog 4	09/05	09/06
Upload meeting minutes 4 & backlog 4	09/08	09/08
First draft: System requirement specif...	09/02	09/05
First review: System requirement spe...	09/06	09/06
First draft: Quality plan	09/02	09/05
Meeting 5 [SYSTEM-LEVEL DESIGN ...	**09/06/21**	**09/14/21**
Completed first draft of SRS & Quality...	09/06	09/06
Second draft: System requirement sp...	09/07	09/12
Second draft: Quality plan	09/07	09/12
Complete meeting minutes 5	09/06	09/07
Review meeting minutes 5	09/13	09/13
Complete meeting backlog 5	09/08	09/09
Upload meeting minutes 5 & backlog 5	09/14	09/14
Application Skeleton [SYSTEM-LEVE...	**09/08/21**	**09/12/21**
Familiarising with development fram...	09/08	09/08
Build skeleton of web application	09/09	09/12
Build skeleton of backend	09/09	09/12
Search for machine learning model	09/09	09/10
Integrate model into prototype	09/11	09/12
Meeting 6 [SYSTEM-LEVEL DESIGN ...	**09/13/21**	**09/17/21**
Completed skeleton of application & ...	09/13	09/13
Second review: System requirement ...	09/13	09/13



Validation of prototype in target envir...	09/13	09/13
Third draft: Quality plan	09/14	09/14
Third draft: System requirement speci...	09/14	09/15
Complete meeting minutes 6	09/13	09/14
Review meeting minutes 6	09/16	09/16
Complete meeting backlog 6	09/15	09/16
Upload meeting minutes 6 & backlog 6	09/17	09/17
Meeting 7 (Lab 3) [SYSTEM-LEVEL ...	**09/16/21**	**09/21/21**
Review and upload: System requirem...	09/16	09/16
Uploaded SRS & Quality Plan	09/16	09/16
Complete meeting minutes 7	09/16	09/17
Review meeting minutes 7	09/20	09/20
Complete meeting back log 7	09/18	09/19
Upload meeting minutes 7 & backlog 7	09/21	09/21
First draft: Project plan	09/16	09/19
First draft: Risk management	09/16	09/19
First Prototype [REFINEMENT]	**09/16/21**	**09/20/21**
First refinement: Frontend	09/16	09/19
First refinement: Backend	09/16	09/19
Review and finalize best practices for...	09/20	09/20
Meeting 8 [SYSTEM-LEVEL DESIGN ...	**09/20/21**	**09/28/21**
Completed first draft of Project Plan &...	09/20	09/20
Review: First prototype, Project plan ...	09/20	09/21
Complete meeting minutes 8	09/20	09/21
Review meeting minutes 8	09/27	09/27
Complete meeting backlog 8	09/22	09/23
Upload meeting minutes 8 & backlog 8	09/28	09/28
Second draft: Project plan	09/20	09/26
Second draft: Risk management	09/20	09/26
Second Prototype [REFINEMENT]	**09/20/21**	**09/26/21**
Second refinement: Frontend	09/20	09/26
Second refinement: Backend	09/20	09/26
Meeting 9 [SYSTEM-LEVEL DESIGN ...	**09/27/21**	**10/01/21**
Completed second draft of Project Pla...	09/27	09/27
Complete meeting minutes 9	09/27	09/28
Review meeting minutes 9	09/30	09/30
Complete meeting backlog 9	09/29	09/30
Upload meeting minutes 9 & backlog 9	10/01	10/01
Review: Second prototype, Project pl...	09/27	09/27
Third draft: Project plan	09/27	09/29
Third draft: Risk management	09/27	09/29
Third Prototype [REFINEMENT]	**09/27/21**	**09/29/21**
Third refinement: Frontend	09/27	09/29
Third refinement: Backend	09/27	09/29
Meeting 10 (Lab 4) [SYSTEM-LEVEL...	**09/30/21**	**10/05/21**
Upload prototype, project plan & risk...	09/30	09/30
Completed third draft of Project Plan ...	09/30	09/30
Complete meeting minutes 10	09/30	10/01
Review meeting minutes 10	10/04	10/04
Complete meeting backlog 10	10/02	10/03
Upload meeting minutes 10 & backlog...	10/05	10/05
First draft: Design report (Software m...	09/30	10/03
First draft: Configuration Management...	09/30	10/03



Completed first draft of Design Report...	10/03	10/03
Meeting 11 [SYSTEM-LEVEL DESIGN...	**10/04/21**	**10/12/21**
Complete meeting minutes 11	10/04	10/05
Review meeting minutes 11	10/11	10/11
Complete meeting backlog 11	10/06	10/07
Upload meeting minutes 11 & backlog...	10/12	10/12
Review: Design Report & Configuratio...	10/04	10/04
First draft: Change Management Plan	10/04	10/10
First draft: Release Plan	10/04	10/10
Second draft: Design report (Software...	10/04	10/10
Second draft: Configuration Manage...	10/04	10/10
Completed first draft of Change Mana...	10/11	10/11
Formulate Test plans	10/04	10/10
Meeting 12 [TESTING + PRODUCTI...	**10/11/21**	**10/15/21**
Review: Change Management Plan, R...	10/11	10/11
Complete meeting minutes 12	10/11	10/12
Review meeting minutes 12	10/14	10/14
Complete meeting backlog 12	10/13	10/14
Upload meeting minutes 12 & backlog...	10/15	10/15
Upload: Change Management Plan, Re...	10/12	10/12
Uploaded Change Management Plan, ...	10/12	10/12
Prepare presentation slides	10/12	10/13
All testing completed	10/15	10/15
Conduct final testing and refinement	10/13	10/15
Meeting 13 (Lab 5) [PRODUCTION]	**10/14/21**	**10/15/21**
Application presentation	10/14	10/14
Upload Test Reports	10/14	10/15
Upload Peer Review	10/14	10/15
Upload CMMI Level 2 Definition	10/14	10/15
Final documents review	10/14	10/15
Final Project Submission	10/15	10/15

5. Operational Requirements

5.1 Help Desk Support

Users have round the clock access to getting help from the help desk support. The help desk support email is readily available and easily found on the Help and Support page. Users may send in an email anytime should they encounter any technical issues or require assistance for issues such as slow system response, inaccessible features or account lock-out. All feedback is welcome too. Users can expect a reply within 3 working days.

5.2 Application Services and Technical support

KCMC training model will be maintained on a nightly schedule, to ensure that the recommended movies are most relevant and up to date with user's preference. The release of the newly trained model will first be tested with a small number of accounts before its full launch.

The network administration and database administrator are required to keep logs on activities on the system and closely monitor for any abnormal activities. They are the front liners in ensuring consistent and high uptime. Should any unexpected downtime occur, the development team would be first informed. They should have adequate knowledge to react promptly and return the system back to its best state.

5.3 Administration Features

System security and access levels are provided in KCMC. There are 2 levels of system access and functional authority, user and admin. Each user is limited to his/her own booking records and the main database of movies, cinema and shows. Admin has access to all user records and the ability to read, write and update the database. Therefore, the admin is responsible for providing adequate support to users when necessary.

5.4 System hardware fail over and routine back up

KCMC runs on user's computers and mobile phones. KCMC aims to be easily accessed through all browsers and electronic devices. The user interface will be maintained to ensure consistent design throughout all screen sizes available on the market.

Backup routines are scheduled to be carried out on a fixed date bi-weekly by the development team, this prevents any loss of user data in the event of a system failure.

5.5 Audit Trail

All booking activities will be stored in the database. A record will include details such as movie, showtime, seat number(s), number of tickets.

Account details will also be stored, which includes username and password.

6. Functional Requirements

Kim's Convenience Modern Cinema (KCMC) is a movie booking web application which provides personalised movie recommendations with the help of deep learning to create a recommendation system.

6.1 User Self-service

Users can create their own account and log in to book a movie anytime. Among others, the online booking system will have the following functionalities:

6.1.1 Account Management

- Register an account with email and password
- Reset/Forgot password
- View booking history

6.1.2 Browse Movie

- View movie poster, cast, genre, duration, release date

6.1.3 Book A Movie

- Select available showtime
- Choose seat
- Complete booking process

6.2 Movie Recommender System

KCMC believes in creating a user experience that will seek to improve retention rate, which in turn translates to savings on customer acquisition through mainstream advertisements.

6.2.1 Training Model with Deep Learning

- Model takes in users' booking history to return recommended movies based on genres

6.2.2 Scheduled Training of Model

- Train model with new data input (eg. booking history, new release)
- Keep recommendation updated to user's latest preference

7. Non-Functional Requirements

7.1 Reliability

- KCMC should maintain an uptime of 98% to eliminate inconvenience brought to users.
- Important data should be backed up regularly (bi-weekly) to prevent loss of data.
- The development team should test in a virtual environment before any release of new features/functionalities.
- The Quality Assurance team should routinely (weekly) conduct tests for bugs.

7.2 Usability

- KCMC UI will strive for consistency by using the same layout and color theme.
- KCMC must allow for easy reversal of actions through the use of a back button.
- Users should be able to complete purchases with little to no trouble, with the use of buttons, within 10 clicks.

7.3 Performance

- System should be able to handle up to 100 users concurrently.
- KCMC UI must keep response time within 3 seconds, to minimise latency.

8. Input Requirements

8.1 User access

All users must create an account before using KCMC. The user must login to KCMC with their username and password before being granted access to the booking system. This allows the system to correctly identify the user and recommend movies personalised to their booking history, therefore the choice of username must be unique in the database.

To prevent tampering of the database, admins would also be required to log in with their credentials. This also acts as a form of security for the users on KCMC.

8.2 Booking details

For users to navigate through the system and successfully book a movie, they would be required to input their choice of movie, showtime, cinema location and choice of seats. The choice of movie will be saved into the database and used for the training of the deep learning model.

9. Process Requirements

The following are among the inherent requirements that KCMC must be able to handle.

9.1 Database transaction

The system must be able to store and receive user data and movie data from our SQLite database system.

9.2 Data integrity

When a database transaction is committed, it should be atomic. The correct results should be reflected, else it should support rolling back the changes if the commit happens to be timed-out or incompleted.

9.3 Data validation

Users’ login credentials will be validated with the stored credentials, any incorrect credentials will result in a failure in logging in.

Throughout the booking process, users are only allowed to click on buttons (eg. movies, seats, showtime), this allows for a low chance of invalid data since the format is fixed and consistent.

The results of the trained model’s accuracy can be validated through the model accuracy, the hit rate percentage of the recommendation model. The higher hit rate percentage signifies the model is able to accurately predict the user’s movie preference.

9.4 Performance

KCMC user interface response time should be kept within 3 seconds.

KCMC should be able to handle at least 100 users concurrently. The admin and development team will be immediately informed when a locking situation occurs. The admin is also responsible for responding to all incoming email so clear any user's enquiry or doubt in the fastest time possible.

The model is trained nightly to ensure high performance at providing an accurate recommendation.

9.5 Data repository

The main repository of data for KCMC is SQLite. SQLite is a self-contained, file-based SQL database. It comes bundled with Django and allows for a seamless development without having to install any additional software. This data repository should be maintained throughout the whole lifecycle of the project. The admin is responsible for adding data such as movies, movie details, cinemas, showtimes, etc into SQLite via the admin portal. The users' details such as username and passwords, and inputs such as booking history will also be stored in SQLite. Lastly, the model will retrieve the training data from SQLite, and update the recommended movies for each user.

10. Output Requirements

10.1 Transaction summary and confirmation

Each user must have a view of a summary of actions done for a particular movie booking session. The database will store all successful transactions and be able to display them to the users if needed. The user will be able to see the title, show time and cinema of the booking.

10.2 Email Service

Email will be used to contact the users. This is used for a variety of reasons, such as for booking confirmation or password recovery.

10.3 Model Accuracy

The cinema admins can view a report of the movie recommendation model's accuracy by accessing the admin page of the webapp. The report will also be stored in the database for future reference.

11. Hardware Requirements

Our system consists of two main components: web application and backend server. Each component requires a different platform to operate and data is transferred through the public internet.

11.1 Network

Both movie goers and cinema admins need the Internet connection to use this system. Movie goers can use any devices that have an Internet connection to load the web application in the browsers. With regards to cinema admins, a computer with Internet connection is recommended to connect to the management website.

11.2 Client Devices

The web application is developed to work well on Google Chrome, Firefox, Microsoft Edge and Safari. It is recommended to access our websites through such browsers.

11.3 Backend servers

For the demo, our server is deployed in a Virtual Machine provided by the School of Computer Science and Engineering in NTU. Here are the specifications of the server:

Component	**Specification**
RAM	8GB
CPU	2 core CPUS
Disk	50 GB

We deployed both SQLite databases and Django server on this virtual machine. As the virtual machine is maintained by NTU IT team, it is well-maintained and guaranteed high availability.

For future deployment, the cinemas will use their own servers, which minimum specification is the same as the demo server.

12. Software Requirements

12.1 Client Operating Systems

- MacOS
- Linux(any distribution with GUI)
- Windows
- Mobile phone / Tablet

12.2 Client Application

The web application must be run on JavaScript compatible browsers including:

- Firefox
- Chrome
- Safari
- Microsoft Edge

12.3 Network system

Network software and protocols used to ensure communication between systems:

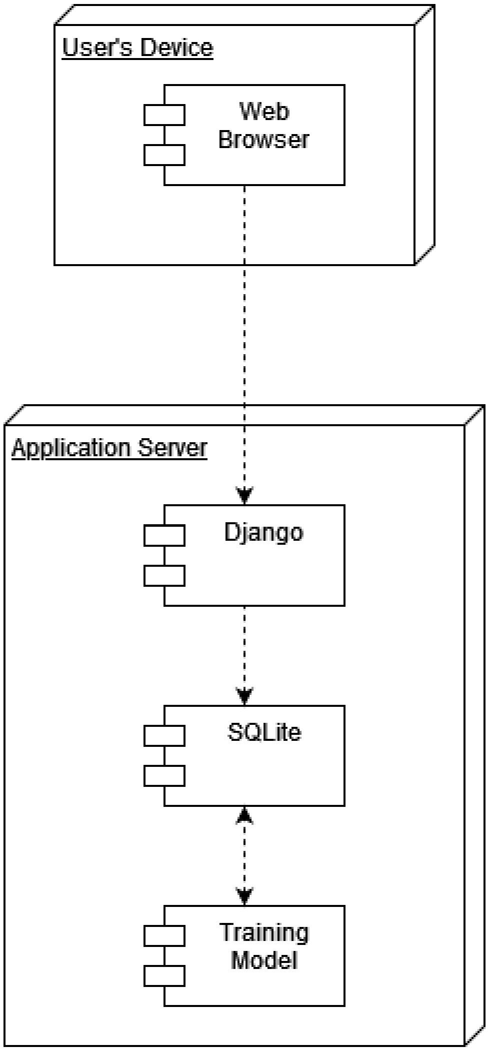
- TCP/IP
- HTTP
- HTTPS

12.4 Licenses

Valid licenses are required to run software from third party vendors:

- Django is an open source project, it doesn’t require any license in development or production.
- SQLite is in the public domain and does not require a license.

13. Deployment Requirements



References

[1] INFOCOMM MEDIA DEVELOPMENT AUTHORITY, “OVERVIEW OF SINGAPORE CINEMA 2020.” SINGAPORE, Singapore, 31-Dec-2020.

[2] Statista, “Number of visits to cinemas in Singapore from 2011 to 2020,” https://www.statista.com/statistics/1020951/cinema-attendance-numbers-singapore/ (Singapore, June 2020).

[3] Towards Data Science, “Deep Dive into Netflix’s Recommender System,” *https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48* (David Chong, 30 Apr 2020).