Kim's Convenience Modern Cinema

[Project Plan]
Version 1.4

By: Wong Ying, Hussain Khozema Kheriwala, Jeremy Book, Desmond Yap, Ta Anh Duc, Taneja Parthsarthi

Revision History

Revision Number	Date	Primary Author(s)	Comments
1.0	28 Sep 2021	Wong Ying	Added 1. Introduction 2. Project Organisation 3. Process Definition 4. Project Schedule
1.1	28 Sep 2021	Ta Anh Duc	Added 1. Product Checklist 2. Best Practice Checklist 3. Quality Assurance 4. Monitoring & Control
1.2	29 Sep 2021	Wong Ying, Ta Anh Duc	Added Project Estimates
1.3	29 Sep 2021	Taneja Parthasarthi	Added Risk Management
1.4	01 Oct 2021	Wong Ying, Hussain Khozema Kheriwala, Jeremy Book, Desmond Yap, Ta Anh Duc, Taneja Parthsarthi	Finalised Version

Table of Contents

1 Introduction

1.1 Project Overview

Kim's Convenience Modern Cinema (KCMC) is a movie booking web application integrated with a recommendation system. The recommendation system aims to provide personalised movie recommendations with the help of machine learning. 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.

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.

1.2 Project Description and Scope

KCMC is a new project being undertaken by Team Kim Convenience's client. Team Kim Convenience has been contracted to fulfill the planning and development of the software, and the client has agreed to deploy the use of KCMC onto local cinema chain's websites.

The basis of KCMC is movie booking web application using Python's Django web development framework with an added feature of a movie recommender system based on the user's booking history. It takes several possible constraints into account. These can be such limits as:

- The user has a registered account on the booking platform.
- The user has an existing booking history.
- The user would like to watch movies in similar genres as his past bookings.

These items are given as an example. Final possible constraints will be discussed in a later document.

The following are the primary features of this application:

1. **Intuitive and Clean Movie Booking Platform**

   For the ease of usage and portability(desktop web, mobile web, etc.), the user interface will be based on simplicity. To promote a simple and fuss-free booking process, the interface should be intuitive with usual layout, having a navigation bar at the top of the page. The navigation bar should allow users to easily access main pages such as Home, Movies and Login

2. **Recommender system - Targeted Advertisement to the right audience**

   KCMC will reproduce Netflix and Disney+ row-based recommender systems. Using such a system will be familiar with users as it is coherent when presented with a row of items that are similar. This is also beneficial for the companies as this helps to collect informative feedback, where a right-scroll on the row reflects an interest in the recommendations while a scroll-down (disregarding the row) would suggest non-interest.

2 Project Organization

2.1 Team Structure

The following is the list of executive roles, as required by CMM level 3.

- Senior Management: Wong Ying
- Software Configuration Manager: Desmond Yap
- Software Engineering Project Group: Hussain Khozema Kheriwala, Jeremy Book
- Software Quality Assurance Engineer: Taneja Parthsarthi
- Representative of Customer: Taneja Parthsarthi, Ta Anh Duc

2.2 Roles and Responsibilities

Project Manager: Wong Ying

- Represents the team to external clients/Keep stakeholders informed
- Plans and oversees project progress
- Prepare budgets
- Approves and executes project plan
- Manages and motivates team members
- Allocates tasks and reports project status to team members

Lead Developer: Hussain Kheriwala

- Overall technical lead for the project
- Translate business requirements into technical requirements
- Directs the development team in the design, development, coding, testing and debugging of applications
- Mentoring team members
- Adhere to determined software quality standards

Front-End Developer: Desmond Yap Qing Yang

- Produces, maintains and modifies websites and user interfaces
- Creates tools that enhance the user's website experience
- Ensures websites are accessible across many platforms, including laptops and smartphones
- Fixes any website issues or bugs that arise
- Works effectively with back-end developer and lead developer

Back-End Developer: Jeremy Book

- Write clean code to develop functional web applications
- Troubleshoot and debug applications
- Collaborate with Front-end developer to integrate user-facing elements with server side logic
- Designs and implements data storage solutions
- Works effectively with front-end developer and lead developer
- Ensures stability and response time of the system meet the requirements

Release Engineer/Manager: Ta Anh Duc

- Builds and Oversees release platform
- Access Software Performance
- Set release schedule
- Ensures application releases are delivered on time and within budget
- Make use of Version Control Systems to coordinate release content and effort
- Builds and integrates changes for delivery

Quality Assurance Engineer/Manager: Taneja Parthsarthi

- Supervises overall product and process quality
- Responsible for creating quality documentation and reports
- Ensures software meets quality benchmark and business requirements
- Designs testing strategies
- Executes test procedure
- Analyse test results

2.3 Team Communication

Team KCMC communication channels include the following:

- Fortnightly physical meetings are held on Thursdays.
- Fortnightly virtual calls discussions are held on Tuesdays and when needed.
- Group announcements and updates are sent through Telegram text channel
- File sharing facilitated through Google Drive cloud storage and the team's Media Wiki page
- Source code control facilitated via GitHub
- Split into 2 key subgroups: Documentation Team and Development Team to work more efficiently.

3 Process Definition

3.1 Lifecycle Model

Team Kim's Convenience intends to use the Object Oriented Evolutionary Prototyping model. This model is especially applicable for this project as it supports exploratory programming through the combination of iterative and incremental models.

The main advantage is that the client's confidence increases as he constantly gets quantifiable goods or services from the beginning of the project to verify and validate his requirements. The model allows for changing requirements as well as all the work to be broken down into maintainable work chunks.

The first prototype should be ready within 3 weeks of finalising the basic design. This ensures a comfortable timeline for debugging and refinement after feedback has been provided by the users on the product for the planning stage of the next cycle. This allows the development team to respond by improving the product, evolving KCMC over time.

The prototype produced will be implemented with a limited set of functions of the movie booking platform. It aims to serve as a proof of concept with fully functional basic and important features, however it is by no means the final product to be delivered.

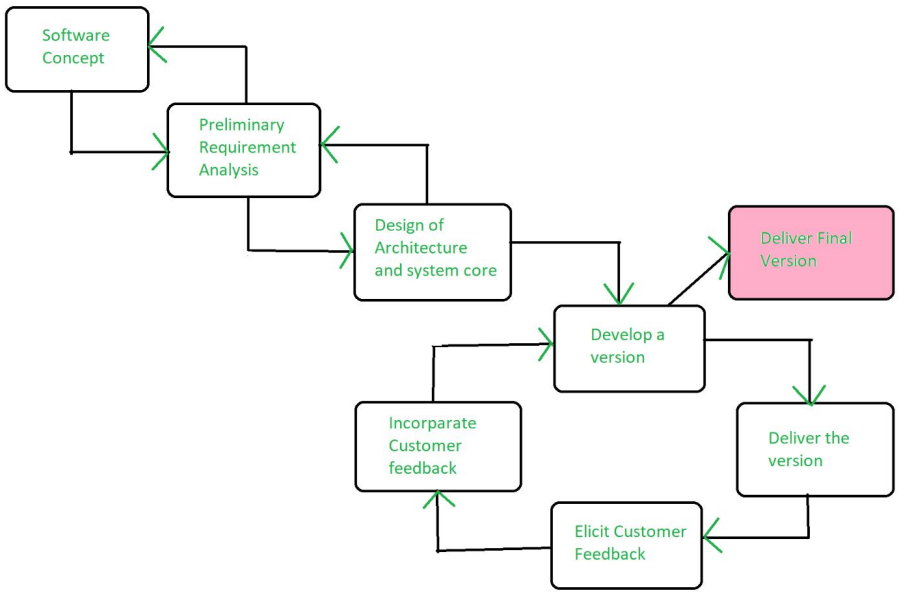

Figure 1: Evolutionary Model Structure

As KCMC aims to provide accurate recommendations and a great user experience, the user feedback phase will be placed with great importance. This is for further system enhancements, including design of user interface and a better recommendation model. After each feedbacking phase, the prototype is refined if it is seen as insufficient to include responses from the users, where the team will make further iterations until it is deemed satisfactory.

4 Project Schedule

4.1 Activity Dependencies and 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.

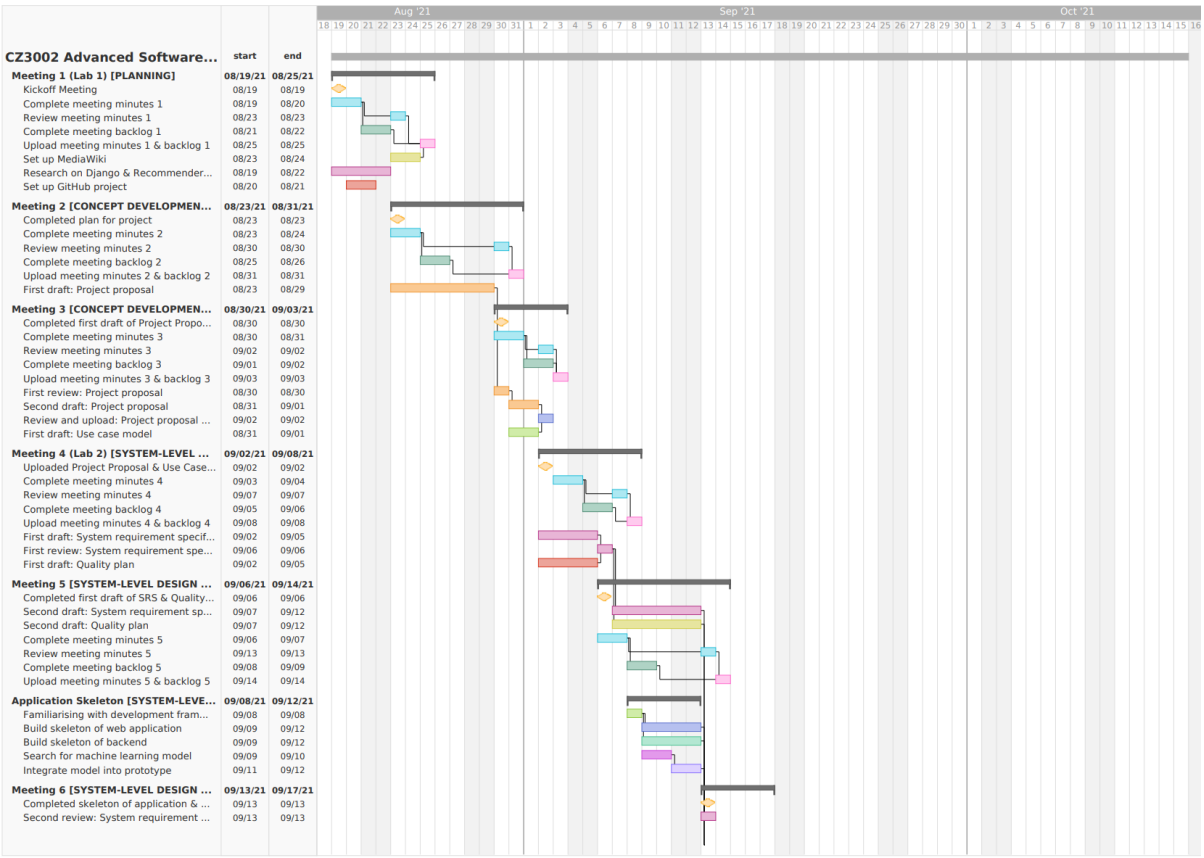

Figure 2: Gantt Chart Part 1/3

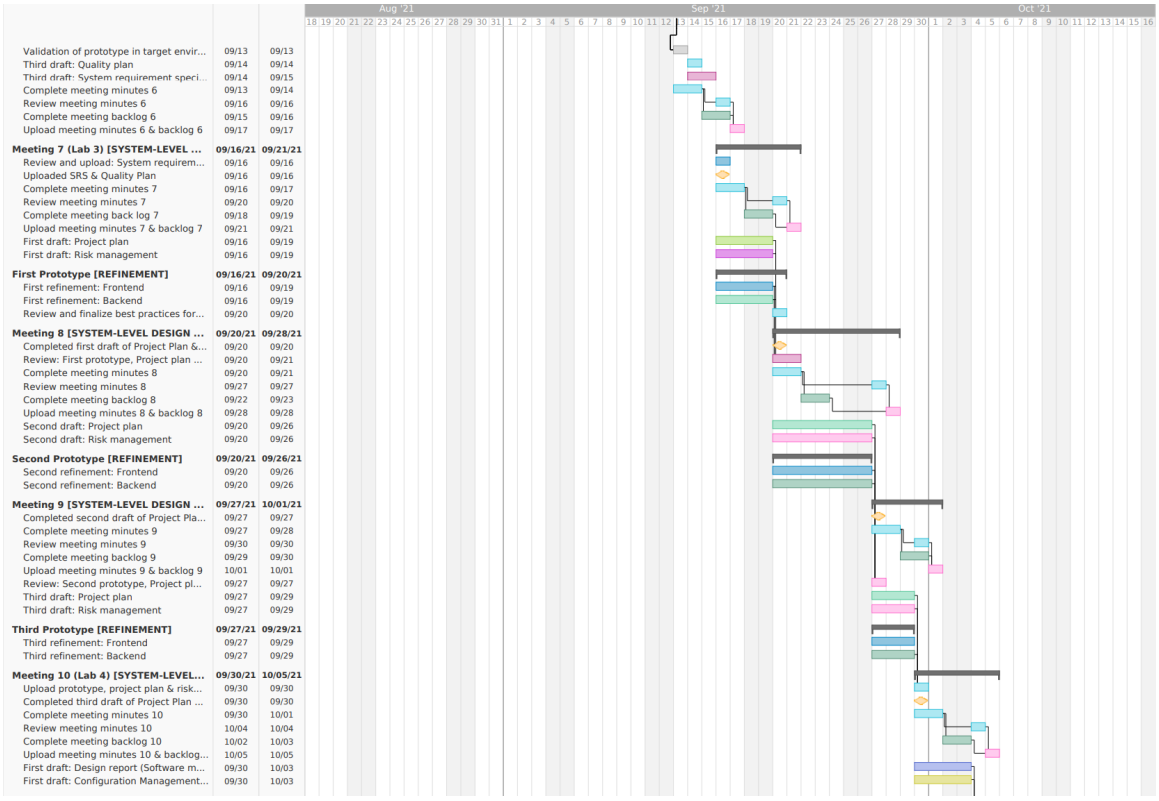

Figure 3: Gantt Chart Part 2/3

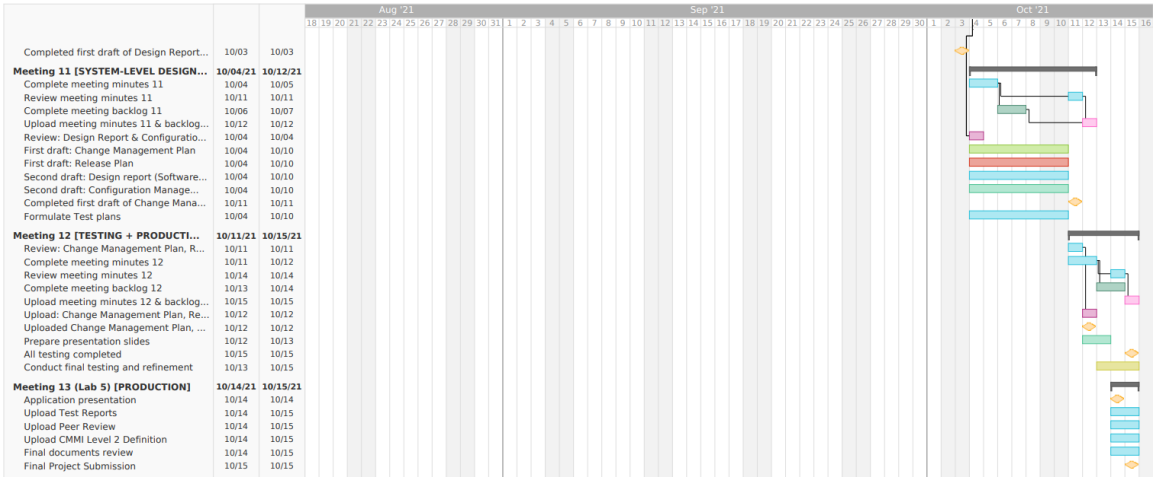

Figure 4: Gantt Chart Part 3/3

4.2 Work Breakdown Structure

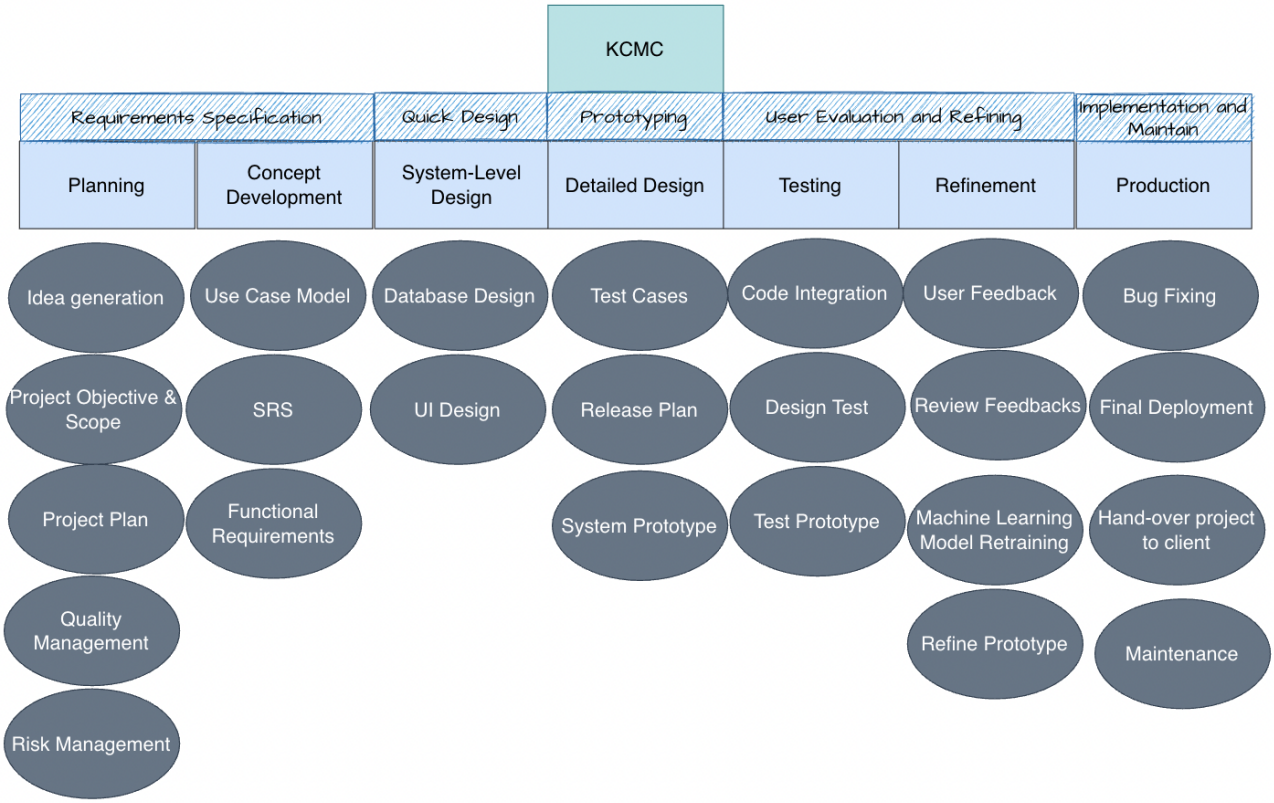

Figure 5: Work Breakdown Structure

4.3 Work Packages

The entire project work is broken down by the important phases of the software development life cycle. They include the following:

1. Project Plan
2. Requirement/Concept Specification
3. Software Architecture Design
4. User Interface Design
5. Machine Learning Model
6. Prototyping
7. Evaluation and Refining
8. Integration and Testing
9. Deployment and Maintenance

4.4 Activity Dependencies

The following table describes the dependencies of the deliverable work packages:

Work Package #	Work Package Description	Duration	Dependencies
X01	Project Plan	14 days	--
X02	Requirement/Concept Specification	14 days	X01
X03	Software Architecture Design	14 days	X02
X04	User Interface Design	14 days	X02
X05	Machine Learning Model	14 days	X02
X06	Prototyping	21 days	X03,X04,X05
X07	Evaluation and Refining	7 days	X07
X08	Integration and Testing	7 days	X08
X09	Deployment and Maintenance	15 days	X09

The following Activity Network Diagram describes the above in more graphical detail:

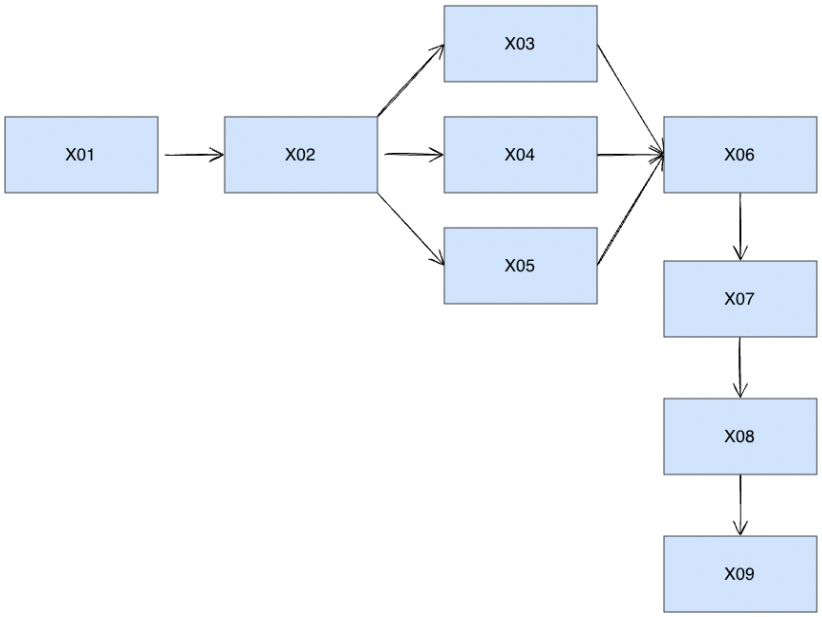

Figure 6: Activity Dependency

4.5 Work Package Details

Work packages are listed below. A team member, indicated in bold, has been assigned as primarily responsible for each work package and will coordinate that package.

Project	KCMC
Work Package	X01 — Project Plan (1 of 9)
Assigned To	**Wong Ying**, Hussain Khozema Kheriwala, Jeremy Book, Desmond Yap, Ta Anh Duc, Taneja Parthsarthi
Effort	14 PD
Start Date	19 Aug 2021
Purpose	To formulate an introductory overview of the project, to be refined in subsequent work packages when documenting the requirements of the project.
Inputs	None
Activities	This work package includes providing a brief overview of the project, its objectives, and a set of proposed project deliverables throughout the development of the software cycle. The people responsible for this work package will also be transcribing ideas brought up in the group meeting discussion into a formal report. It also inspects client's requirements and build requirements.
Outputs	A written document of the Project Proposal, Project Plan, Quality Management, Risk Management, Scheduling & Budget Management

Project	KCMC
Work Package	X02 — Requirement/Concept Specification (2 of 9)
Assigned To	**Wong Ying**, Hussain Khozema Kheriwala, Jeremy Book, Desmond Yap, Ta Anh Duc, Taneja Parthsarthi
Effort	14 PD
Start Date	02 September 2021
Purpose	To establish a clear, detailed and common understanding between the client and Team Kim's Convenience of the client's requirements to be addressed by the project
Inputs	X01 — Project Plan
Activities	Identifying and interviewing targeted audiences to understand their needs, formally documenting and inspecting the client's requirements.
Outputs	A written document of the Use Case Model, System Requirement Specifications and Functional Requirements

Project	KCMC
Work Package	X03 — Software Architecture Design (3 of 9)
Assigned To	**Hussain Khozema Kheriwala**, Desmond Yap, Jeremy Book, Wong Ying, Ta Anh Duc, Taneja Parthsarthi
Effort	14 PD
Start Date	16 September 2021
Purpose	To do design high-level architecture of the system
Inputs	X02 — Requirement/Concept Specification
Activities	Various components in the system should be identified in the high-level architecture design, showing their inter-relationship and how they interact with one another. Other details such as the software and hardware infrastructure, choice of operating system, choice of language for implementation should also be decided. Design topics such as maintainability, portability and reusability should also be kept in mind.
Outputs	High-Level Design and Architectural Specification

Project	KCMC
Work Package	X04 — User Interface Design (4 of 9)
Assigned To	**Desmond Yap**, Hussain Khozema Kheriwala, Jeremy Book, Wong Ying, Ta Anh Duc, Taneja Parthsarthi
Effort	14 PD
Start Date	16 September 2021
Purpose	To develop a user interface with high level of usability, allowing the user to interact with the system
Inputs	X02 — Requirement/Concept Specification
Activities	Revision of low-fidelity user interface prototype with client to help with early visualization of design. This ensures less confusion and miscommunication before moving ahead with the high-fidelity user interface prototype.
Outputs	Low-Fidelity User Interface Design, High-Fidelity User Interface Design

Project	KCMC
Work Package	X05 — Machine Learning Model (5 of 9)
Assigned To	**Jeremy Book**, Desmond Yap, Hussain Khozema Kheriwala, Wong Ying, Ta Anh Duc, Taneja Parthsarthi
Effort	14 PD
Start Date	16 September 2021
Purpose	To generate a model which can produce recommendations with a high level of accuracy.
Inputs	X02 — Requirement/Concept Specification
Activities	Develop a machine learning model which learns from a booking history database. The accuracy achieved should be at least 80%.
Outputs	Machine learning model, List of recommendations for each user

Project	KCMC
Work Package	X06 — Prototyping (6 of 9)
Assigned To	**Hussain Khozema Kheriwala, Jeremy Book, Desmond Yap**, Wong Ying, Ta Anh Duc, Taneja Parthsarthi
Effort	21 PD
Start Date	23 September 2021
Purpose	To produce a prototype of a system with basic and important functionalities for the client. This allows for an initial evaluation of the work done before further increment.
Inputs	X03 — Software Architecture Design, X04 — User Interface Design, X05 — Machine Learning Model
Activities	Development team will implement all the requirements gathered into a prototype. Some basic functionalities that are expected to be included are: User account creation and login, Movie recommendation system, Movie and Show browsing, Movie booking, Editing of Movie and Show details
Outputs	Source code for application and working user interface (prototype) with basic functionalities of KCMC

Project	KCMC
Work Package	X07 — Evaluation and Refining (7 of 9)
Assigned To	**Ta Anh Duc**, Jeremy Book, Desmond Yap, Hussain Khozema Kheriwala, Wong Ying, Taneja Parthsarthi
Effort	7 PD
Start Date	7 October 2021
Purpose	To encourage feedback and suggestions from the client when the prototype is presented to them . This process might be iterative until the client is satisfied with the product.
Inputs	X06 — Prototyping
Activities	Presentation of prototype to client. Client giving their opinions and feedback for improvement. The development team will refine the prototype accordingly to release a new prototype. This process is repeated until satisfaction has been achieved.
Outputs	Evaluation reports, Updated requirements elicitation

Project	KCMC
Work Package	X08 — Integration and Testing (8 of 9)
Assigned To	**Taneja Parthsarthi**, Jeremy Book, Desmond Yap, Hussain Khozema Kheriwala, Wong Ying, Ta Anh Duc
Effort	7 PD
Start Date	14 October 2021
Purpose	To identify and fix logical and syntax errors produced during the implementation of the system as well as finalising the system before deployment. To ensure software satisfies requirements stated by the client and meets his standard of quality.
Inputs	X07 — Evaluation and Refining
Activities	Black-box testing is conducted to verify functionality of the system without internal code structure. White-box testing used to verify internal system workings via input-output flow. User testing will be conducted to improve interface design.
Outputs	Completed final system, Test reports

Project	KCMC
Work Package	X09 — Deployment and Maintenance (9 of 9)
Assigned To	**Hussain Khozema Kheriwala**, Wong Ying, Jeremy Book, Desmond Yap, Ta Anh Duc, Taneja Parthsarthi
Effort	15 PD
Start Date	21 October 2021
Purpose	To deploy KCMC, proper handover to client and facilitate maintenance of system.
Inputs	X08 — Integration and Testing
Activities	Ensure timely development, meeting the agreed deadline by securing resources and developing deployment strategy plans. Upon deployment, continuous effort on improving and debugging of the system should still be put in.
Outputs	SVN repository, smooth deployment and proper maintenance of system.

5 Project Estimates

5.1 Code Size Estimation using Function Points

We calculated unadjusted function point based on the complexity of functions provided by this system. Code size is then estimated by adjusted function point.

5.1.1 Unadjusted Function Points

KCMC supports the following proposed functions:

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

The measure of unadjusted function points is based on five primary component elements of these functions: Inputs, Outputs, Inquiries, Logical Files, and Interfaces. Each element ranges from Low Complexity, Medium Complexity to High Complexity. The detailed evaluation of the complexity is as follows:

Rating Inputs:

- Register new account (i.e. username,email address, password and phone number)
- Book a movie ticket (i.e. choice of movie, showtime, cinema location and choice of seats)
- Add a movie (i.e. title, id, cast, director, language, run length, certification, popular index, trailer and image)
- Add a showtime for movie (i.e. movie, screen, date and time)

Files Type Referenced (FTR)	**Data Elements**		
	1-4	5-15	Greater than 15
Less than 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
Greater than 2	Average (4)	High (6)	High (6)

Rating Outputs:

- Transaction summary and confirmation (i.e. movie title, show time and cinema)
- Email Service to contact customers
- Model Accuracy of machine learning model

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 3	Average (5)	High (7)	High (7)

Rating Inquiries:

- User login onto KCMC
- Displaying the movies according the customer's criteria
- Selecting customer booking
- Displaying of recommended movies for customer

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (3)	Low (3)	Average (4)
2 or 3	Low (3)	Average (4)	High (6)
Greater than 3	Average (4)	High (6)	High (6)

Rating Logical Files:

- User account detail
- Movie details
- Movie showtimes
- Seating arrangement
- Movie booking

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low(7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Rating Interfaces:

- Register account: Email user authentication

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low(7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Summary of above analysis:

Element	Complexity	Detail
Inputs	Medium	Register new account
	Low	Book a movie ticket
	High	Add a movie
	Medium	Add a showtime for movie
Outputs	Medium	Transaction summary and confirmation
	Low	Email Service to contact customers
	High	Model Accuracy of machine learning model
Inquiries	Low	User login onto KCMC
	Medium	Displaying the movies according the customer's criteria
	Medium	Displaying of recommended movies for customer
	Low	Selecting customer booking
Logical Files	Low	User account details
	Medium	Movies details
	Low	Movie showtimes
	Low	Seating arrangement
	Low	Movie booking
Interfaces	Low	Register account: Email user authentication

Calculation of Unadjusted Function Points:

Characteristic	Low		Medium		High	
Inputs	1	× 3	2	× 4	1	× 6
Outputs	1	× 4	1	× 5	1	× 7
Inquiries	2	× 3	2	× 4	0	× 6
Logical Files	4	× 7	1	× 10	0	× 15
Interfaces	1	× 5	0	× 7	0	× 10
Unadjusted FP	46		31		13	
Total=L+M+H	90					

5.1.2 Adjusted Function Points

Influence Factors	Score	Detail
Data Communications	5	Data will be transferred through many components in the system
Distributed Functions	3	Distributed processing and data transfer are online and in both directions.
Performance	4	Response time is critical at all time. The data should be processed within a few seconds to ensure user's satisfaction.
Heavily used	5	The website will be used by more than 100 users everyday
Transaction rate	5	The volume of data transaction is estimated to be around 1000 entries per day
On-line data entry	5	More than 50% of transactions are interactive data entry
End-user efficiency	3	Four to five of the efficiency designs are included
On-line data update	4	Database update is needed for account registration, managing users' bookings and providing movie recommendations.
Complex processing	4	The system utilises a state-of-the-art machine learning model to provide movie recommendation to the users according to their booking and movies' rating.
Reusability	5	The system is structured and documented for ease of re-use. The source code and documentation are available on Github.
Installation Ease	3	The step-by-step installation document is provided on Github to help with the setup of the system.
Operational Ease	5	The system is designed to focus on ease of use for the users, so it must be user-friendly
Multiple sites	3	The system will be deployed to more than one installation site.
Facilitate change	3	Changes within the system need to be done on time and accurately to ensure smooth operation.
Total score	57	
Influence Multiplier = Total score × 0.01 + 0.65 = 57 × 0.01 + 0.65 = 1.22		
Adjusted FP = Unadjusted FP × Influence Multiplier = 90 × 1.22 = 109.8		

Scoring (0 – 5)
0 = No influence
1 = Insignificant influence
2 = Moderate influence
3 = Average influence
4 = Significant influence
5 = Strong influence

5.1.3 Lines of Code

According to Capers Jones statistics, each Function Point requires 39 lines of code if the application is implemented using JavaScript.

Therefore, we have: **Lines of Code** = 109.8 FP × 39 LOC/FP = **4283 LOC**

5.2 Efforts, Duration and Team Size Estimation

To estimate the effort and duration required for the project, we use function points as the basis to calculate Effort, Duration, Team size and finally the schedule. The estimates are expanded to account for project management and extra contingency time to obtain the total average effort estimates. From these averages, the duration of each work package in working days is estimated based on the following calculations.

- Working days include 5 days in a week.
- Effort = Size / Production Rate = (4283 LOC) / (39 LOC/PD)[1] = 110 PD
- Duration = 3 × (Effort) $^{1/3}$ = 3 × (110) $^{1/3}$ = 14.4 Days
- Initial schedule = 14.4 Days / 5 days a week = 2.88 Weeks
- Team size = 110 PD / 14.4 D = 7.64 P = 8 Persons
- Working hours include 8 hours in a working day.
- Total person-hours (PH) = 110 PD × 8 hours = 880 PH

5.2.1 Distribution of Effort

1990's Industry Data	Work Package	Distribution	Estimates
Preliminary Design 18 %	Project Plan	9%	79.2
	Requirement Specification	9%	79.2
Detailed Design 25 %	User Interface	7%	61.6
	Technical Architecture	11%	96.8
	Data Modeling	7%	61.6
Code & Unit Testing 26 %	Code & Unit testing	21%	184.8
	Online Documentation	5%	44
Integration & Test 31 %	Integration & Quality Assurance	31%	272.8
	Extrapolated total effort		880
	2% for project management		17.6
	3% for contingency		26.4
	Total effort		924

These duration estimates are based on the assumption that each team member works an equal amount on any given work package.

[1] Lines of code per Person Day statistics based on Industrial Benchmarks, 1997: 31 LOC/PD for United States; 62 LOC/PD for Canada

5.3 Cost Estimates

Category	Project Expenditures	Price	Quantity	Total
Hardware	Computers	$1000/unit	6	$6000
Software	SQLite	$6000 (one time fee)	1	$6000
	Private Git Repository	$24/month	4	$96
	Development SDK (Oracle)	$5000 (one time fee)	1	$5000
	Domain and Hosting	$779.52/year	1	$779.52
Manpower	Project Manager	$6000/month	4	$24000
	Lead Developer	$4800/month	4	$19200
	Front-End Developer	$4250/month	4	$17000
	Back-End Developer	$4250/month	4	$17000
	QA Engineer /Manager	$4600/month	4	$18400
	Release Engineer /Manager	$4250/month	4	$17000
Miscellaneous	Printing, Paper, Photocopy, Transport	$1000	-	$1000
	Rental	$450/month	4	$1800
			Implementation cost:	$133275.52
Contingency	Contingency	10% of Implementation costs	-	$13327.55
Maintenance	Private Git Repository	$24/month	12	$288
			Total:	$146891.07

6 Product Checklist

The plan is that the items listed below will be delivered on the stated deadlines.

Project Deliverable	Estimated Deadline
Project Proposal	02 Sep 2021
Use Case Model	02 Sep 2021
System Requirements Specification	16 Sep 2021
Quality Plan	16 Sep 2021
Project Plan	07 Oct 2021
Risk Management Plan	07 Oct 2021
Prototype	07 Oct 2021
Design Report on Software Maintainability	21 Oct 2021
Configuration Management Plan	21 Oct 2021
Change Management Plan	21 Oct 2021
Release Plan	21 Oct 2021
Presentation	21 Oct 2021
Test Plan	28 Oct 2021
Test Cases & Requirements Test Coverage Report	28 Oct 2021
CMMI Level 2 Definition	28 Oct 2021
Final Product	28 Oct 2021

7 Best Practice Checklist

Practice	
Every work should be documented along the development progress so that all team members are in sync and aware of the changes made by others. The documentation should be standardised and organised in a systematic manner.	
Standard practices for requirement specification: ● Specification should be verifiable, traceable and unambiguous ● Term should be defined clearly and used consistently ● Must contain a complete functional specification	
Pay attention to complexity management: ● Minimize interfaces between modules, procedures and data ● Avoid doing more on requirements and ensure all functionalities developed meet customers' requirements.	
Software change management ● New features must be branched out from baseline	
There should be transparency in the team such that: ● The manager can monitor the progress. ● Avoid overlapping work by making sure each team member is aware of each other's responsibilities ● Good communication amongst members ● Source codes and system design are available to review to find software defects	
Project should be continuously updated to improve: ● All manuals design, test, source code should have revision numbers, dates, revision history comments and change marks to indicate the changes. ● New revisions should be approved during planning and checked for quality and compliance after being made. ● Documentation should be updated whenever there is a change.	
A good estimation should be made at the start of the project to avoid underestimating the required time and effort, especially on integration, testing, documentation and maintenance.	
Software testing including black box, white box testing: ● Generate test cases that covers all possible scenarios ● Involve the unit, functional, integrating and acceptance testing ● Include stress tests.	

8 Risk Management

A risk is an event or condition that, if it occurs, could have a positive or negative effect on a project's objectives. Risk Management is the process of identifying, assessing, responding to, monitoring, and reporting risks.

Kim's Convenience Modern Cinema will employ a proactive risk management strategy. This is accomplished via careful and continuous evaluation of critical project areas in order to identify risk events, work as a cohesive team to determine the potential impact and damage of each identified risk, develop and implement mitigation and avoidance strategies for each identified risk, and monitor the risks until they are resolved.The project manager working with the project team and project sponsors will ensure that risks are actively identified, analyzed, and managed throughout the life of the project. Risk identification will involve the project team, appropriate stakeholders, and will include an evaluation of environmental factors, organizational culture and the project management plan including the project scope.All risks identified will be assessed to identify the range of possible project outcomes. For each risk that will be mitigated, the project team will identify ways to prevent the risk from occurring or reduce its impact or probability of occurring.The level of risk on a project will be tracked, monitored and reported throughout the project lifecycle.

A Risk Log will be maintained by the project manager and will be reviewed as a standing agenda item for project team meetings.The following table is an excerpt from Risk Management Log:

Risk Type	**Risk Description**	**Occurrence**	**Severity**	**Priority**	**Mitigation / Contingency Plan**
People	Key personnel unavailable during time critical period	High	Serious	High	Get available personnel (i.e. working on low priority tasks) that understands task at hand to temporarily take over key personnel role
Tools	Possibility of single point failures, such as the application or the KCMC movie recommendation model malfunctions	Medium	Serious	High	Have modularity and portability in code design and dependencies so that modules can be independent shifted around

Technology	Software and hardware components used for development are not compatible (for example, choosing a incomprehensive tech stack)	Medium	Catastrophic	High	Research extensively prior on the compatibility issues between components to be used as well as to ensure portability
Organizational	Restructuring of organization such that separate teams (roles defined in the Project Proposal and further division of sub teams) are now working on the system	Low	Tolerable	Medium	Use of source code management and version control via Github for software control. Clear division of work and frequent check-ins (via text/call/internal presentations) for documentation.
Estimation	Inaccurate estimations and deadlines that are either overly optimistic or unrealistic	High	Tolerable	Low	Break the work down into smaller pieces/activities then perform estimation for a more accurate and realistic estimation through thorough discussions.

Risk Management is discussed further in a separate document, namely Risk Management Plan.

9 Quality Assurance

The project will achieve quality assurance by following the standard set by the team at the beginning of the project. The specific procedures and details shall be provided in the Quality Plan.

Specific test procedures and details shall be provided in the Module/System Test Plan.

In addition, our project shall make use of four testing methodologies:

- **Unit Testing**: involves the testing system components individually. This testing should be done in different phases to ensure that software bugs in each component are found in the early stage.
- **In-Place Testing**: involves testing of the whole system as a unit to make sure that all modules are integrated properly and produce the expected results.
- **Performance Testing**: involves testing of the system's performance under a particular scenario. The system's performance will be measured in terms of responsiveness and stability. This testing will help to ensure that the system can perform well when there is a large number of concurrent users. Furthermore, the recommendation model will also undergo testing to ensure the accuracy of more than 80%.
- **User Acceptance Testing**: involves testing of the system from the user side. This testing will help to ensure that the system is user-friendly and can meet users' expectations.

Furthermore, these methodologies will be used to test two important aspects of KCMC:

- **System Function** will be tested to ensure that software flaws are eliminated, and
- **Algorithmic Function** will be tested to ensure that heuristic aspects of the project (such as user preference rankings) perform realistically to provide value to the users.

KCMC's methodology makes broad use of realistic test cases. Comprehensive and detailed dummy data will be provided for the testing purposes to ensure that the system behaves correctly. Moreover, realistic scenarios will be used to validate code in many different aspects and extreme cases will also be considered.

10 Monitoring & Control

Many procedures are required in order to be able to successfully monitor the progress of a software project. Some of the most important are:

Quantitative measurement of resource consumption: Estimates of the KCMC's resource requirements, primarily in terms of human resources, can provide a quantitative measurement of project progress when compared to progress in terms of project milestones. The percentage estimates of each milestone's resource requirements provided in this document allow for easy progress tracking.

Identification of major project risks: Early identification of major risks to the project allows for placement of preventative measures before problems can develop. Major risks have been identified in the Risk Management section of this document, along with the measures being taken to avoid them.

Regular reviews of project progress: Throughout the duration of the KCMC project, the team shall meet weekly to review the progress of all project tasks, including management, planning, analysis, development, and testing. This allows us to identify problems early on and ensure the project's milestones will be met on time. The main communication point is the Telegram group, where we used to discuss problems and arrange time for the weekly meeting. Github is used to share the codes amongst team members and keep track of the changes in development. Relevant documentations and their drafts such as SRS or project plan will be created and shared on Google Drive for ease of access. After receiving approval from the whole team, the documentation will be stored on the Team Wiki.

Timeline Planning and task decomposition: This document outlines an estimated timeline for the project. A reasonably accurate timeline can be assembled by hierarchically decomposing tasks into measurable subcomponents and estimating requirements for each. At the same time, this decomposition can assist in task assignment and balancing. Throughout the implementation phase, these subcomponents can allow for fine-grained measurement of progress. Project subcomponents and timeline estimates are included in the Schedule, Estimates and Work Breakdown Structure sections of this document.