University of Plymouth

School of Engineering, Computing, and Mathematics

PRCO304
Final Stage Computing Project
2019/2020

Rico Assistant

Billy-Ray Haggar

10561789

BSc (Hons) Computer Science

Acknowledgements

I would like to thank my project supervisor, Lingfen Sun, for her support and guidance throughout this project. I would also like to thank Mario Gianni for his knowledge and guidance on recommender systems. Large thanks go to Marco Palomino, my personal tutor, for his support with all personal problems and enthusiasm with reminders of my capabilities. This extends to David Walker for the knowledge he shared throughout my degree, and the sincere help by going above and beyond to help me achieve my best.

Finally, I would like to thank my partner, Holly, for all the late nights, support and encouragement during my whole degree. She kept me on track and pushed me to work harder all whist she completed her own studies.

Abstract

This report covers the software development project of ‘Rico Assistant’. Rico is an application that utilizes a new, simple recommendation system in order to suggest and manage items to users of the software. Rico Assistant is a framework setup to grow and recommend items in all manners of areas.

The report covers the initial background of recommendation systems and their uses today with their current limitations, along with how these can be improved through a different approach. Next, legal, social, ethical and professional regulations are discussed with how Rico Assistant keeps on top of these areas. From here on, an overview of how Rico Assistants project management approaches and technologies are used in an effective development cycle is covered.

The main body discusses the designs, system architecture and development of Rico Assistant, along with an outline and discussion of design principles, plans and software technologies used. Continued with sprint by sprint conclusions of each week of development with the main ideas being discussed in further detail. Testing results and discussions come next to conclude the development cycle thus far; leading to this report.

The final 3 sections conclude this report, all evaluating this project with is achievements, short comings and evaluations of design choices. Analysis into planned ideas and developed ideas are present with a critical evaluation of effectiveness with the end user.

Finally find a reference, bibliography list and an appendix, with all related documents to this report are attached.

Contents

WORD COUNT: 9995

GitHub Code Submission Link (Invite Only):
https://github.com/BillyHaggar/Rico_Assistant

OneDrive Code Submission Link:
https://liveplymouthac-my.sharepoint.com/:u:/g/personal/billy-ray_haggar_students_plymouth_ac_uk/EYW91Eoe5zNBl8j3o1D_4sUBra6qPzhLwVrSNF-pbYw9sA?e=P1Fv1F

Deployed Project (Testing Hidden): https://rico-assistant.herokuapp.com

1. Introduction

Current recommendation systems (also known as recommender systems/engines) are vital in the technology world of today. With the exponential growth of online data leading to the world of big data, the need for recommendation systems is on the increase to solve the problem of filtering such large datasets [*Lü et al., 2012].* By filtering such large data, results for a targeted end user/viewer can gather results best optimised for themselves.

Currently, many companies implement recommendation systems in order to recommended items to boost sale of items, improve social aspects such as recommending new friends and connections, and also for the targeted consumption of media such as music and movies [*Deng, 2019; Lü et al., 2012*]. This allows the end user to have data/items suggested that will most likely be of a match, resulting in better profits for businesses. Amazon states with targeted recommendations, 20% - 40% of all profits are from what they consider 'worst sellers' [*Lü et al., 2012*], as by targeting users personalised products that would never have been considered before, these items are now brought to the front and centre of the ideal customer, who would be most likely to purchase.

The problem with these current recommendation systems are that they don't fully base their intent of use on the end user, it results more in how they can boost sales/views. Furthermore, many recommendation systems require large datasets and many inputs from a user in order to provide accurate recommendations, this is called a 'cold start' [*Lü et al., 2012*]. There are also further problems with recommendation systems such as scalability, diversity and accuracy, evaluation of recommendations and the user interface of the system.

'Rico Assistant' aims to re-design the implementation of a recommendation system, by fully focusing on the end user and their needs; deciding activities to do or movies to watch in order to cure their 'boredom', removing the need to decide for themselves. Rico also aims to solve some of the major problems in current recommendation systems such as cold starts, scalability and evaluation of recommendations. The whole software package aims to be as simple and as reliable as possible as to make this form of recommendations in the grasp of all; no over complications, just what is required. Rico is fully responsive as to be available to all desktop or mobile users no matter the device.

2. Background

2.1 What is a Recommendation System?

Recommendation systems automate the search for best suited data, they aim to predict ratings for items that a specific user may give, without the user giving a rating to that specific item. They also filter out irrelevant data for the aimed user through this rating of items. Recommendation system ideology is to analyse existing data (user rating histories) to generate the recommendation predictions [*Jafarkarimi et al,. 2012*]. Several approaches to recommendation systems are present, however, the most common is 'collaborative filtering' [*Breese et al. 2013*], where the system assumes that what users have agreed on in the past will more than likely be similar to what is wanted in the future. By using similar users of similar rating histories, predictions of new items can be assumed. This approach allows for recommendations to be made where no actual knowledge of the item to be recommended is needed, for example recommending a movie may need to have context of genre, age and budget but collaborative filtering only requires user interest.

2.2 Problems with Recommendation Systems

As mentioned often recommendation systems require large training datasets in order to accurately recommend items specific to a user which can result in a 'cold start'. This means that a new user to the recommender system, time and input is required in order to give recommendations that are of any form of accuracy, the amount of time and input varies but ultimately, results in the loss of user interest. Due to the initial lack of accurate suggestions/recommendations the user may decide to not continue using the recommendation system [*Bobadilla et al, 2012*], further decreasing the training dataset, further decreasing accuracy.

Sparsity, another weakness results from needing more data to predict recommendations than the number training ratings obtained [*Moreno et al., 2016*]. This relates to cold starts with not holding enough data in order to accurately recommend items. Scalability of recommendation systems propose problems in the form of computational performance, as all data training or recommendations increase, the computational power to classify and compute recommendations increase [*Zhao, Sheng, Zhu and Wang, 2018*].

Ultimately, these are the three main downfalls of collaborative filtering recommendations systems. The main problems that result from the above are false positive and false negative recommendations [*Moreno et al., 2016*], these are what causes a user to lose interest as why use a system that doesn't give the user what they wanted/expected from the software. Could this be solved though a simpler system that always puts the user first? this is what Rico aims to solve.

2.3 Current Implementations / Market Research

Recommendation systems are being increasingly used to target users for a variety of reasons; there are multiple implementations of them. Numerous companies implement recommendation systems for a better user experience and increase of profits from user interaction. By researching what implementations are currently in use, this helps to build upon them to produce a better, simpler system for all.

2.3.1 Netflix

Netflix, a media streaming service, invoke the use of a collaborative recommendation system in order to display best matched movies/tv shows to users of their platform in order to retain user subscriptions. By recommending items that users will most likely want to view, users will be left feeling that keeping a subscription will be beneficial to them. Netflix gains it data through binary means, such as, has a user watched a certain title to completion? Thus, indicating that they were of a 'good match' to the recommendation. Netflix gets past the cold start issue through what they call a 'jump start', where on being a new viewer to Netflix you select items that you are of interest in, giving initial training data that can classify you near users of similar interests [*Netflix, 2020*].

2.3.2 Spotify

Spotify is a music streaming service that has similar reasoning to Netflix as to why to implement a recommendation system. Spotify does not take any initial data, leaving a 'grace period' of letting a user search for the music they listen to naturally at first, thusly providing training data before being recommended items. Recommendations firstly will be given based on music popularity at the time of use, then based on a user specific interests when enough data has been collected. Spotify and its recommendations work differently by suggesting music to a user that is different to said users typical interests, even keeping songs in suggested playlist for up to 4-weeks without user interaction. This results in a higher chance of user interaction with items they wouldn't normally [*Boam, 2019*].

2.3.3 Amazon

Amazons recommendation system for items utilize nearest neighbour AI in order to recommend items on an item-by-item basis [*Deng, 2019*]. By suggesting items to userA from userB's best matched items, where userA and userB are similar in rating history, new accurate recommendations are suggested. This tracks 'worst sellers' and broadens item audience in order to sell more hard to sell items. Amazon found suggesting items based on correlations between items rather than correlations between customers, better link together similar users. Amazon initially found problems in this algorithm as some of these suggestions were considered poor by

the user. The problem was that Amazon needed to assess the likelihood of purchasing itemB (suggestion) based on ‘any’ possible purchase not just itemAs purchase; Amazon stated “*That was a large improvement to recommendations quality, when we got the math right” [Hardesty, 2019].*

3. Legal, Social, Ethical and Professional

3.1 Legal

Complying with legal regulations is vital for any project/software to follow, as to not incur problems with any legal authority. Anything used within the production of software must have explicit use to use within licences. Any material used must have the owner's permission to use or be covered under any rules and regulations that apply to a project of this nature. For example, many materials or licences state in their terms of use that under fair use and for education purposes, items may be used as long as there is no commercialisation of the project. Still, serious efforts are in place for Rico to comply and be as close to releasable software as possible, especially with software now requiring to be more compliant with legal, social and ethical regulations. Constant checks during release of software must be completed in order to confirm legal regulations are still being met.

3.1.1 Licences of Software & Material used

MEAN stack
Ricos software structure, based on a mean stack, is completely open source, meaning that there is no required licence to use and all technologies are considered in the public domain. Being open source allows Rico to use, edit and share any code in relation to the use of the MEAN software stack [*opensource.com, 2020*].

SKmeans Node Package
Rico uses a node package created by another developer, and so checks in regard to legal use were conducted. SKmeans and its dependencies are released under npm with a MIT licence, meaning the usual open source legislation applies. Furthermore, an MIT licence states "*Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions*" [*opensource.org, 2020*]. Under these terms Rico is fully licenced to use SKmeans in order to achieve its K-Means classification.

Heroku
In order to host 'Rico Assistant', the hosting service Heroku is employed. To be certain that my application and any material used by Rico applies within Herokus terms of service, checks of compliance and an agreement is signed; as stated by the terms of service [*Heroku, 2020*].

3.1.2 General Data Protection Regulations (GDPR) & Data Protection Act

Rico, being a software project that relies on the storage of user data, storage and use of data must be in accordance with GDPR laws and the Data Protection Act. It is vital that data supplied by the user is treated with regards to protection of the user and any legal legislations that apply. Both GDPR and the Data Protection Act seem similar, however, GDPR states that users have the right to know of data use and not to be subjected to automated decisions in regard to personal data. Whilst the Data Protection Act covers legitimate grounds for data use and implements safeguards to prevent against individual rights and freedoms [*DPO centre, 2018*].

Rico follows the Data Protection act [*Legislation.Gov.uk, 2018*] by only storing data that is required for a purpose of the software, as well as removing any links between data and user (where links aren't required). Users have the facility to update any personal information, additionally any data entered by a user is validated to confirm accuracy. Further security measures are implemented to ensure encryption and storage of user data is in a secure format, passwords and user IDs are hashed and salted. Any data transferred to the webpage is under HTTPS encryption, with all possible functions only being run on a secure server.

GDPR, as outlined in the UK government released guidance paper [*ICO, 2018*], is about the handling of the publics data, mainly, any data that is identifiable of a user/person. GDPR is outlined and mentioned in the Data Protection Act 2018 [*Legislation.Gov.uk, 2018*]. With being active UK legislation as of 25th May 2018, Rico must adhere to the regulations as set out by GDPR. Any data within Rico is stored in a secure database, with any user having the right to withdraw any personal data within the application, through the deletion of their account; on account deletion any data linked to said user is no longer recoverable. Users are able to email system admins in order to obtain any data that Rico stores of them.

With testing results from user testing of 'Rico Assistant', all data is treated as anonymous, in addition to allowing test users to withdraw any data by emailing the researcher with their unique ID. This is to comply with GDPR and Data Protection Act standards.

3.2 Social & Ethical

With any software that relies on user data to function, social and ethical issues are faced with the collection of said data. All data that is used must be for a purpose, and one issue that can be faced with user data is sharing with third party companies. Rico does not share any data with third parties, and due to the nature of Ricos aim to be simple, only required data is stored and used for a specific purpose. This is to keep the social and ethical use of data under control. With Ricos recommendations

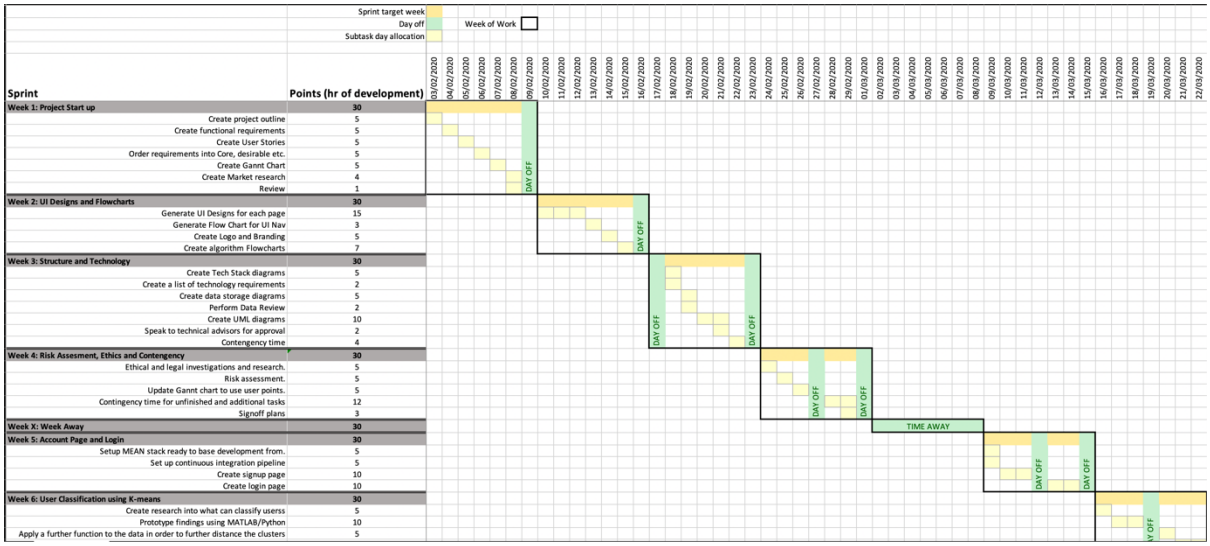
some may not be suitable for that of a certain audience, and so, in plans and implementations of Rico, warnings will be given to cover possible implications to the user if data may not be appropriate. Users are also given control to reclassify if current recommendations are not suitable. Privacy is a large concern for social and ethical issues as proved by the Cambridge Analytica scandal [*Stemeir, 2018*], and so all data is stored securely and sensitive information such as passwords are stored in a manner that it's near impossible to decrypt.

Data collection from public users must also be acceptable in terms of social and ethical issues. All data from test users is treated with anonymity as per the 'University of Plymouth Ethics Policy'. With accordance from the said mentioned ethics policy, all test data was gathered from students of 'University of Plymouth' in accordance with the testing policy. All test users are notified of the use of their data and how they may withdraw if not happy at any point; pre or post completion of the testing.

4. Project Management

Project management is treated with professionalism and respect as to manage work and to ensure at least a minimum viable product is attained. The aim for good software project management is to ensure projects are planned, implemented, monitored and controlled so that any possible weaknesses/failures are planned for and avoided.

Time management is vital in order to ensure that plans are met to an acceptable standard in the time available. Supervisor meetings to review Trello boards and progress ensured acceptable levels of productivity were reached and that the project supervisor approved of progress. This allowed the development of Rico to be checked against schedule (See Figure 1), and if any deviations from plans occurred the project supervisor was available for guidance to keep on track. This goes hand in hand with further assistance from the project supervisor, technical advisors and professional advisors to recommend approaches and technical research. Overall time management and advise allowed Rico to bloom and gain advantages possible from the time available.



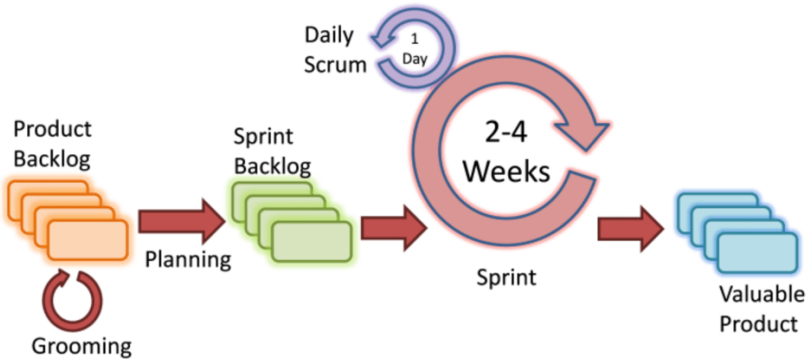
[*Figure 1 – Gannt chart produced to outline Rico Assistants development, this is a rough outline based on each sprint. Each sprint has development points to allocate the set amount of time for development each week.*]

4.1 Method of Approach

4.1.1 Agile Project Management

Agile development was chosen as the choice of development cycle over other methods such as V-Shaped and Spiral methods. Agile development is based on the ideology of delivering value to the end user in small increments (See Figure 2). It's not all about waiting for a software solution that does everything, but rather working

on all aspects at once in order to achieve a minimum viable product; then later working on additional features. Agile works on the idea of sprints, with each sprint being a week or two of development based on specific goals. All these goals go into a product backlog that is a roadmap for what to do for the project at hand. It is a list of jobs developers should refer to at each sprint. Items and sprints are created alongside development as at the start you would work on core functionality and then each sprint creation will set a newer, incremental goal [*Atlassian, 2020*].



[*Figure 2 – Agile management diagram showing the basic steps and flow of constant updates and sprint/product backlog [Sami, 2012]]*

Agile has many benefits over other approaches, such as spiral, but mainly it focuses on end user satisfaction. Agile is also great for products of limited time frames, unlike spiral where there is a high cost and a long time to reach a finished product [*Sami, 2012*]. Furthermore, Agile development gives the ability to respond to changes throughout development, with all plans, requirements and results being evaluated at each stage. This allows for safety in any problems that occur and allows for project supervisor meetings to have a larger effect. In addition, Agile incorporates risk management into each stage of development.

As Initial plans needed to be created to gain an idea of the overall approach and produce the initial backlog, Agile documents such as functional requirements and user stories were given an allotted time. With roughly 3 months for development, elements of the waterfall model were applied to allocate the first month to generate plans required to visualise ‘Rico Assistant’. This allowed for the benefit of verifying plans, ensuring they were clear of errors as well as giving further structure to the project [*Sami, 2012*].

4.1.2 Test-Driven Development.

Agile development benefits greatly from incorporating test-driven development, each stage has tests written in code as to confirm anything developed is functioning correctly. Good test-driven development aims to write tests first set up to fail, then coding the intended function checking it does as expected with the created test. Of

course, the test will fail first, as the code for the function wouldn't have been written yet. This approach assists the developer to write the simplest code in order to pass the test, in effect allowing for good implementation of functions in addition to giving the developer a good understanding on what the code should do. These tests will stay in code for the whole development of the project to ensure that no new development breaks what has already been implemented [*HASELT, 2016*].

4.2 Project Management Technologies

4.2.1 Trello

Trello is a tool used to generate a product backlog for an Agile development project. Trello themselves say "Trello is a collaboration tool that organizes your projects into boards. In one glance, Trello tells you what's being worked on, who's working on what, and where something is in a process." [*Trello, 2020*]. Trello works with Agile ideas to visualise the organisation of sprints and daily scrums. Each sprint has each step to reach its goal as a card on the Rico Assistant Trello board, giving the developers a list of tasks to work on. The organisation of the Trello board starts with outlining the "Sprints to Complete" in one column with each sub task put into the next column "Up Next" moving into "Current Tasks" when development on this aspect must be worked on. When complete, tasks are put into the corresponding column; any bugs discovered during development are put into the "bugs" column as to be worked on when contingency time is available.

Trello was fantastic at visualising and demonstrating to the project supervisor what stage 'Rico Assistant' was at in a glance, this helped at receiving feedback at any meeting. For logs of the Rico Assistant Trello board, see appendix 13.2.

4.2.2 GitHub

All 'Rico Assistant' documents and code required version control with a central storage location for all personnel of the project to be able to access for reference. Utilizing version control allows for development of new concepts to be isolated as not to affect previous additions to software. It also allows for reversions to previous states of code if a certain addition breaks software beyond repair, this also allows for if a previous version is more stable, reversion is quick and easy [Git-scm.com, 2020; *Git Tower, 2020*]. Each commit to the code repository is given a short description as to inform viewers of the repository. This provides knowledge of additions to code, as well as information of any problems, removals and new feature implementations. With each addition to the software solution, a separate 'branch' allowed for each sprint/new feature development to be isolated from the functional master branch.

GitHub served as the version control of choice, mainly due to GitHub integration with other development software such as Travis.CI and Heroku, for a simplified

development pipeline. GitHub acted as an off-site backup of any development files and code, this gave safety in the event of hardware failure resulting in the loss of work. GitHub also allows for easy understanding of each version of software and its current additions/problems due to its commit history.

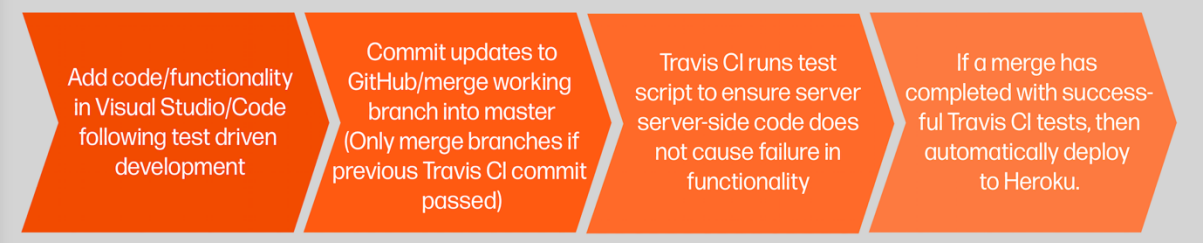
4.3 Sprint Planning

Agile development relies on sprints and their plans, with each sprint being a small incremental goal for the development of the software such as, a user can create an account. "*A sprint is a predetermined timeframe within which the team completes sets of tasks from the backlog" [Littlefield, 2016],* which clearly outlines development goals for each week of development in Rico Assistants case. With the generated 'Sprint Planning Document (Appendix 13.3)', each week a set of goals were placed into the product backlog (Trello) in order to visualise a development plan for developers and supervisors. Each sprint in the document outlines its overall goal, any related functional requirements and user stories, each step to achieve the goal, a timeframe for each sprint and a review to conclude each sprint. This document allows for the project owner to see the overview of development and progression.

With each completed sprint, a review evaluates each stage of development, due to the Agile ideology, adaptions could be made to future sprints as to maximise efficiency in remaining time. See the later section '6. Development' for each sprint related goal and review.

4.4 Development Pipeline (DevOps)

A robust development pipeline gave Rico a constant deployment to its host server with each addition tested for working functionality. If any tests fail, indicating a non-functional release deployment would not occur. The root of the pipeline starts with the generation of code; Visual Studio Code allowed for fantastic management of all code files in a minimal environment. A local Node.js server run on localhost allowed for local developers to view changes live, especially when used in conjunction with Safari's web inspector; linked to an iPhone X. The web inspector let the responsive design of the application to be developed with ease, as code could be edited with a live view on mobile, ensuring that a full responsive design is implemented. As all code was written in a test-driven environment, each function has a relating unit test to be run locally before committing to the source management control.



[*Figure 3 – Diagram of Rico Assistants CI/CD pipeline demonstrating the flow of each stage.]*

The next stage, source management control, GitHub was chosen to hold all code files, this allows for collaboration between developers to develop source code as well as act as a central location for software files (Son, 2019). After each commit to GitHub, integration testing through Travis CI ensures all server-side functions are still functioning as intended, even after additions to the code. Travis CI functions by building the node.js server and running all integrations tests to check if they all pass, if not, a developer is able to view this either through GitHub or Travis CI's dashboard.

The final stage of continuous deployment depends on branch merges to 'master'. After a development branch (based on sprint) is finished, it merges with the master branch, only if all tests have passed from Travis CI. Upon tests passing, and a successful merge to master, automatic deployment to Heroku follows. Rico employs a continuous integration/deployment pipeline bringing its many development benefits along, mainly fault isolation and faster stable releases to the end user [*Katalon, 2019*]. This pipeline also allows for easy additions to the project with everything automated.

5. System Architecture & Design

In-depth planning and research are vital to the success of any project, the plans created should include enough detail as to ensure that anyone viewing the created plans, could recreate a product very close if not the same as the project. Each aspect of the software solution was thought out and plans were drawn up in such a way that all vital functionality is outlined, as well as further additions of later features. Choices into design principles to follow and considerations into possible system architectures confirm that the approach and design for Rico Assistant is the better option. Here are the objectives of the projects, these documents outline what is expected from 'Rico Assistant'.

5.1 Design Principles

5.1.1 Responsive & User Focused Design

Rico Assistant is designed from the ground up to have a one application for all in the form of a responsive web application. No matter the device (Mobile or Desktop), Rico adapts/responds to the users device in order to give a consistent but customised appearance. This all functions based around the user's device screen size to allow a targeted view of the application based on the user's preference. This is the goal of responsive design, to '*eliminate the need for a different design and development phase for each new gadget on the market'* [*Smashing Magazine, 2011*]. Accompanying the responsive design is a simplification of functionality, such as one login for all types of user, Rico will know if you are a general user or an admin and take you to the page customised for you.

The application is user focused at every stage, following HCI principles such as the Shneiderman '8 Golden Rules for user interface design' [*Wong, 2018*]. These rules are applied to Rico to ensure that decent human computer interaction of the application is present, all to ensure that at no stage a user finds difficulty in use of software.

1. Strive for consistency.
2. Enable frequent users to use shortcuts.
3. Offer informative feedback.
4. Design dialogue to yield closure
5. Offer simple error handling.
6. Permit the easy reversal of actions.
7. Support the internal locus of control
8. Reduce short-term memory load.

[*Figure 4 – The 8 Shneiderman rules of interface design*.]

5.1.2 Don't Repeat Yourself (DRY) Principle

The principle of DRY, strives to deliver code that is simple, easy to maintain and reduces the chance of bugs, all while saving time and effort [*Baghel, 2018*]. The DRY ideology is simple, don't repeat yourself and make code as simple as possible using simple ideas. By making the code of Rico Assistant maintainable, any future additions are easy, making each stage of development a little easier to complete. When combining the DRY principle with test driven development, less unit tests need to be written, meaning there is further time saved.

5.1.3 Security

Security of any software application must be considered and applied to protect user data, which is mentioned in the above section of legal investigations. Within Rico all passwords and secure data is encrypted with salts/hashing and transmitted over a secure connection (https), this goes further with protection on each page to ensure that a user is logged in, in addition also checking the user has the correct permissions to access that page. Fields that could be open to a malicious attack are also validated as to only allow data of the correct format.

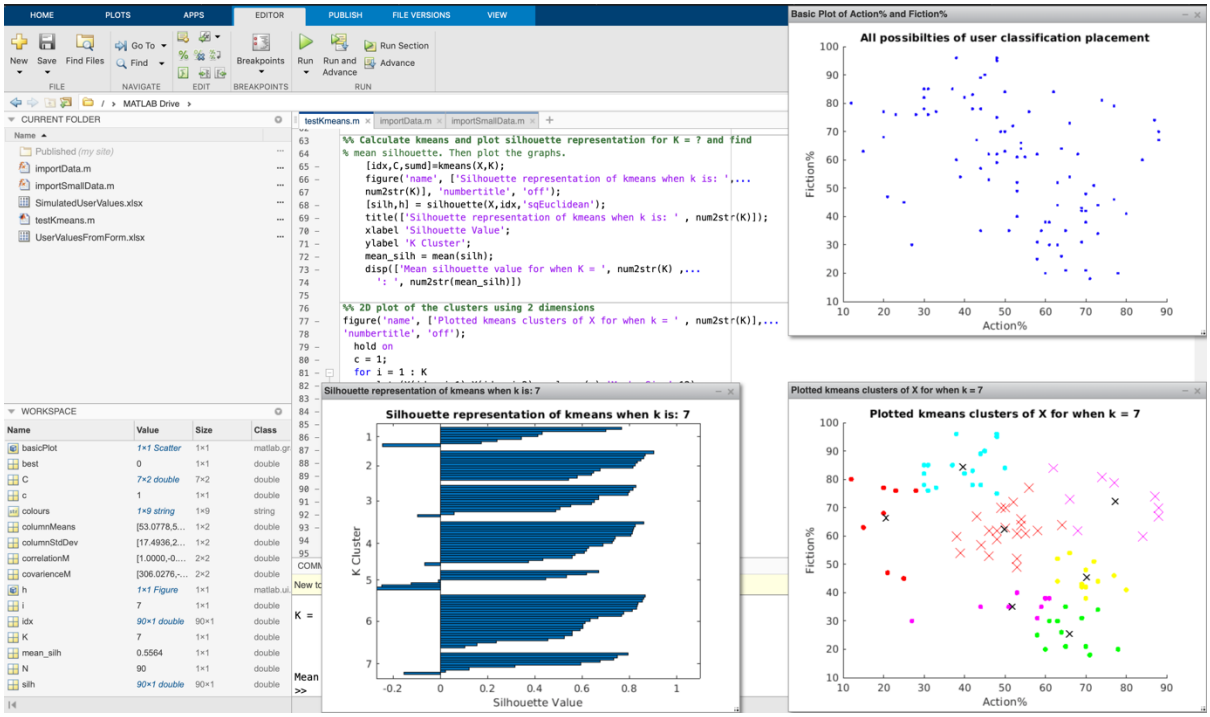
5.2 Design Technologies

5.2.1 MEN Stack

Originally Rico was planned to use a MEAN stack, however, during development it was decided that an Angular app was not required nor the best choice for aspects of Rico. To keep the code simple and easy to understand, a slight edit to a MEN stack seemed more than appropriate. A MEN stack, comprised of MongoDB, Express, and Node.js is tried and tested in professional environments for its reliability and robustness.

5.2.2 MATLAB Prototyping

MATLAB prototyping of classification algorithms was appropriate to ensure that any users of the software will gain accurate recommendations. Mathematical formulas and hence software algorithms are rarely ever produced first time [*Massey, n.d*], and so prototyping of vital algorithms allowed Rico Assistants recommendation system to be guaranteed to give accurate results. Furthermore, this led to simulated test data created from statistical surveys and reports, to be run in both the confirmed working prototype and software implementation; certifying the implementation was a success when compared. For all MATLAB test files refer to the code submission links at the top of the report.



[*Figure 5 – Screenshot of MATLAB prototyping used to tweak and finalise the implementation of a K-means classification system to be used in Rico Assistants Recommender System.]*

5.3 Planning Documents

5.3.1 Functional Requirements

Functional requirements are all items/aspects that a developer must implement into a project in order to reach the end user requirements. These must be made clear and easy to understand. In the case of the functional requirements produced for ‘Rico Assistant’ (See Appendix 13.5), each requirement has been sorted into core, desirable and additional requirements. This is to plan and outline what needs to be focused on for in order to reach a minimum viable product.

5.3.2 User Stories

User stories are based on the functional requirements of the project but with a user centred focus; with these accurate milestones/focuses on development bearing the user in mind, its certain that the software product will fulfil the needs that they require.

Using the outlines of agile project management as set out in ‘User stories Applied’ [*Cohn, 2009*], a solid user story template is explained. These user’s stories should state who the user is, in this case general user or an admin, then go on to state what that user wants and why. With these three areas we can check that the correct user is identified for the requirement, what they want from the software and to check that what was implemented, gives them the result they wanted. Each of these stories, like

the functional requirements, are split up into core, desirable and additional features as to give each task a level of importance. With this document (See Appendix 13.8), along with functional requirements being used for further details, everything that this application requires can be set out.

5.3.3 Non-functional Requirements

From the discussing with Chris Winter; an IT expert and consultant with many years of experience, the idea of non-functional requirements was brought to attention. The idea behind these is that an application and its functional requirements may in fact outline what the software has to perform, for example within a banking application the transfer of funds is a functional requirement. However, an example of a non – functional requirement would be that the transfer itself must be completed within a quick time frame.

The idea of these non-functional requirements is to not only think of what the program/software must do, but also how these functional requirements must be performed. By following the procedure of outlining non-functional requirements we can be certain that yet again we are thinking about the end user, and how they would want an application to perform in correlation to the functionality requested. See Appendix 13.6 for the Non-functional requirements relating to ‘Rico Assistant’.

5.3.4 Misuse Stories

Another topic brought to light by the meeting/discussion with Chris Winter was the idea of ‘Misuse Stories’. The idea here is to cover any case where a user can misuse the software not in the intended purposes. An example of this would be an application to buy food services, and the misuse story would be the user stealing the food items. By covering misuse stories, we can cover any malicious acts that the application may encounter, and thusly build defences within the software solutions to mitigate their impacts or to avoid them entirely. See Appendix 13.7 for the misuse stories relating to ‘Rico Assistant’.

5.3.5 Risk Assessment

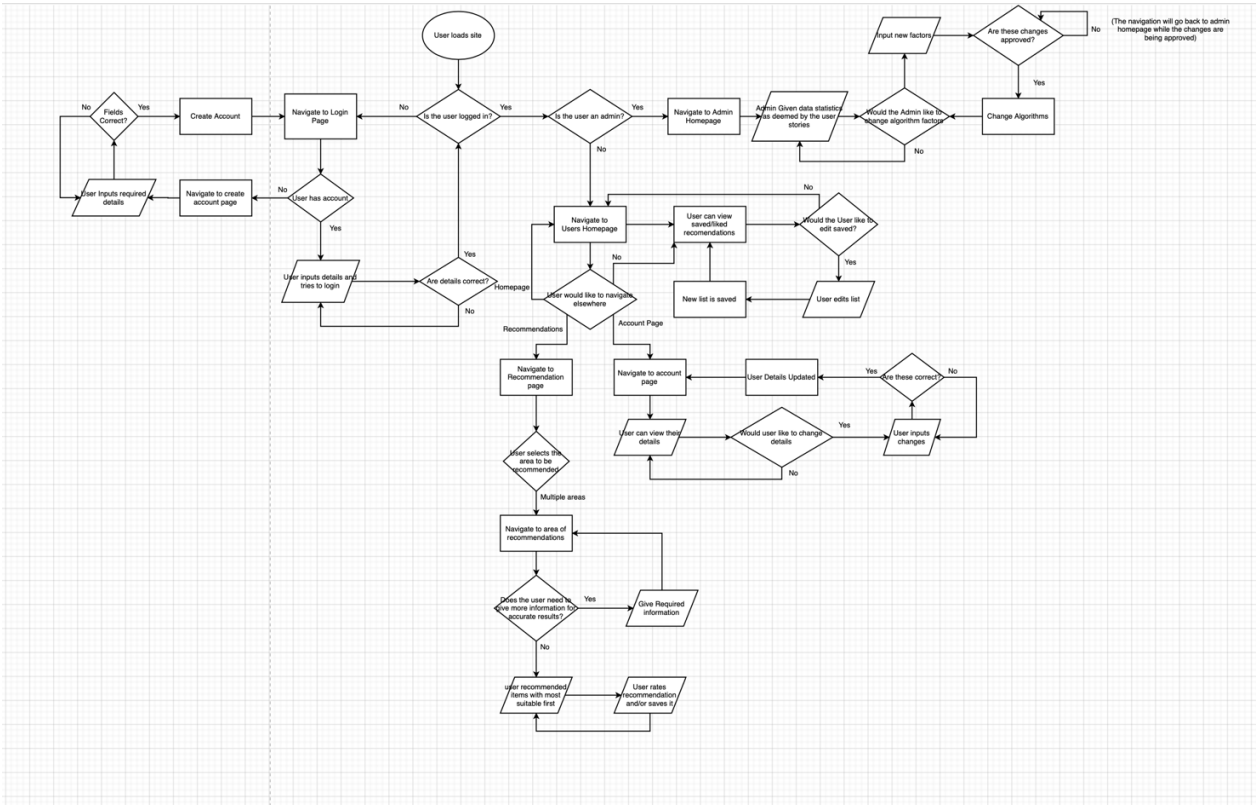
In terms of Agile management, risk assessments are an ongoing investigation and consideration. The traditional approach is to outline any problem/risk in the overall scope of a development cycle, and create prevention techniques along with plans on what to do if an emergency happens and the risk takes place. This technique for the overall project is great at minimising the chance of a main risk but do these handle all risks in relation to the user?

By applying the approach of agile to risks, outlines of risks in relation to each sprint and their relating user stories can be drawn up. At each sprint stage, reviews can occur at each implementation for more direct risk analysis, whilst bearing the end user in mind. Each directed risk is added to the overall risk plan as to allow its discovery to assist future development. For details on Agile risk management and how to layout the risk assessment plan, a journal paper by (Moran, 2016) covers the ideas of applying risk management in an agile manner. With the knowledge and suggestions from this journal, in addition to the above ideas, the concept of incorporating both the traditional method and directed agile methods seems the best of both worlds.

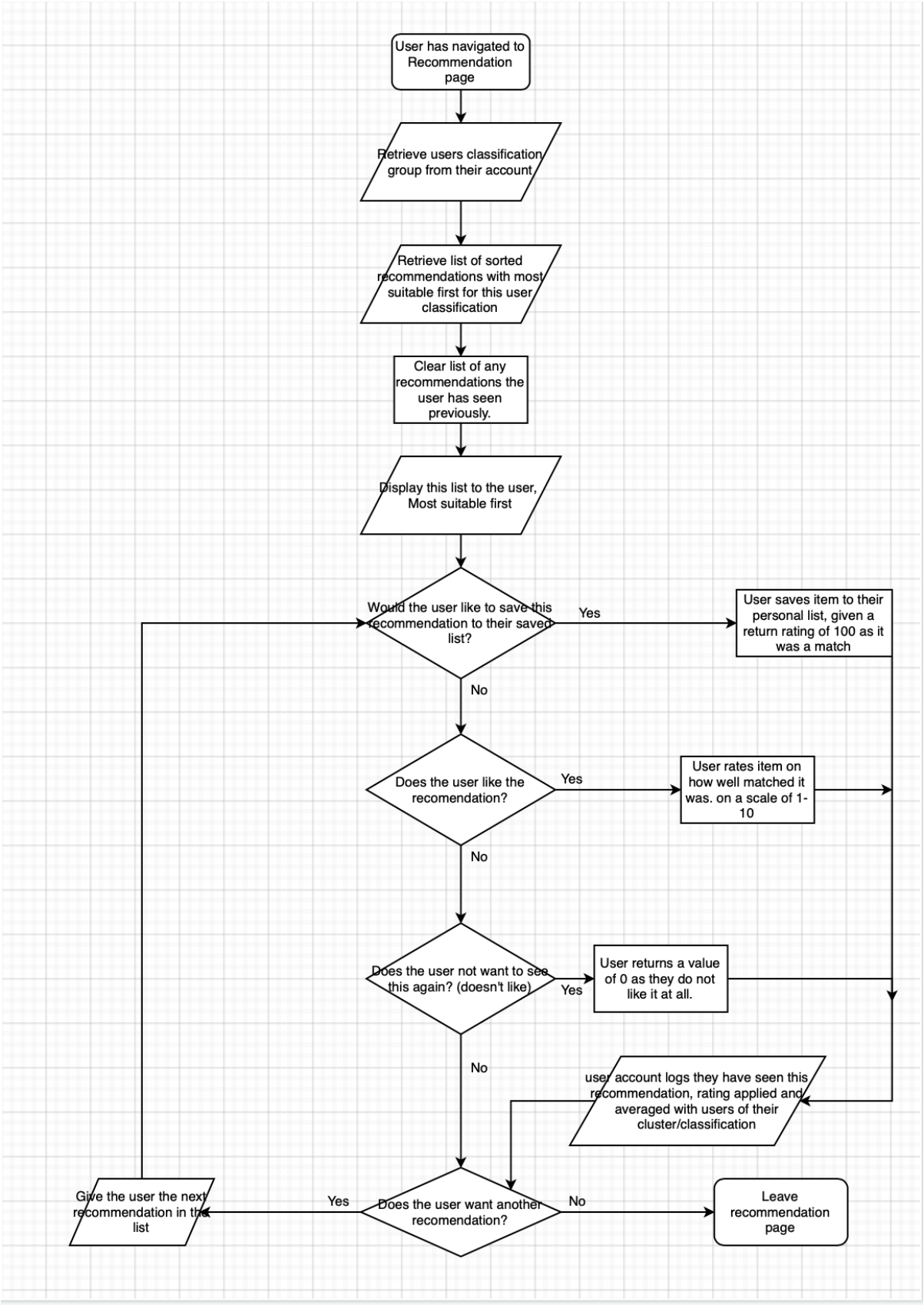
By using the above-mentioned journal entry, three areas (Risk Identification, Risk Monitoring, Risk Treatment) are mentioned in order to apply agile methodology. However, we will use these areas for the traditional method outlining the broader risks as well as the later discovered risks. We shall also give each risk a severity rating as to give an indication to a sense of urgency and importance for each risk. This will allow developers/engineers to know what to focus on in an agile environment. To see the Risk Assessment plan for ‘Rico Assistant’, see appendix 13.9.

5.3.6 Design Diagrams

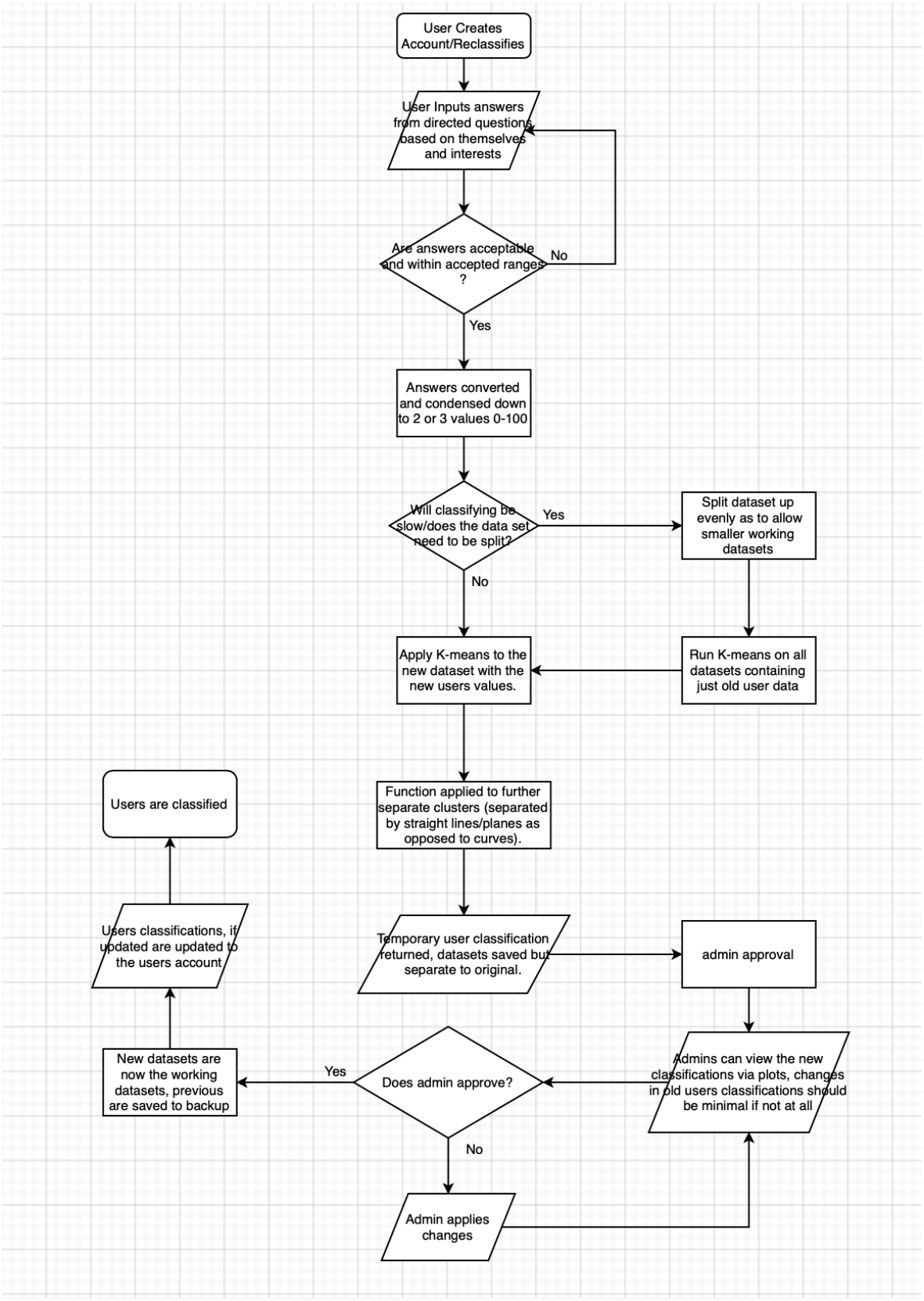
UML diagrams show visual representation of the intended implementation of a system. With the produced diagrams for ‘Rico Assistant’, smaller aspects of the intended system can be visualised. Sections such as UI-Flow (Figure 6) and Algorithms (Figures 7-8) can be represented with ‘Flow Charts’, with these being a great way of showing the decision logic that make up these aspects. These Flowcharts can be designed around user stories as to ensure that these are met in the finished software solution. Full size versions of all UML diagrams can be found with the code submission.



[*Figure 6 – Basic UI Flow of Software Pages*]

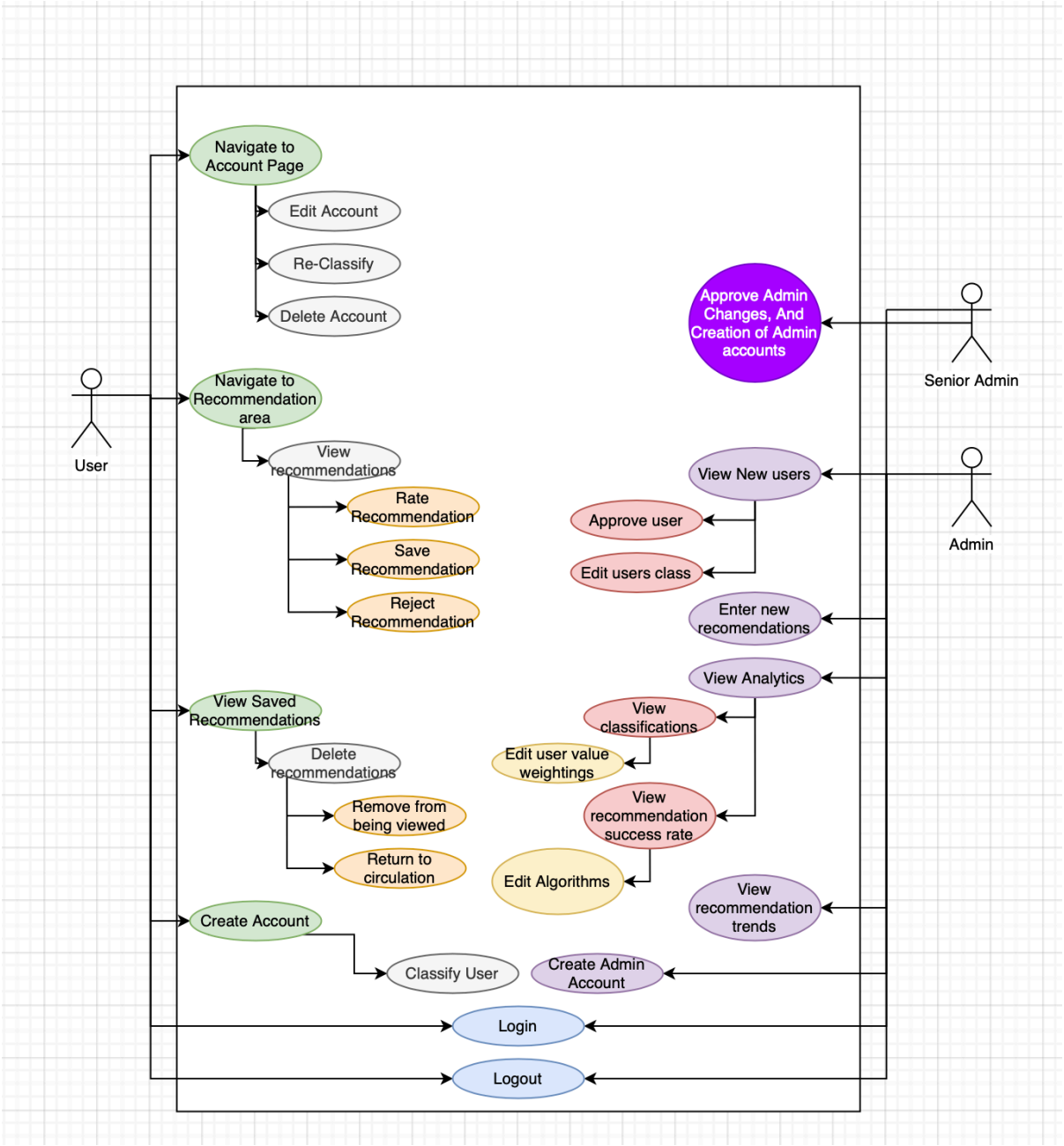


[*Figure 7 – Flowchart demonstrating the algorithm for the recommendations to be displayed to a user.]*



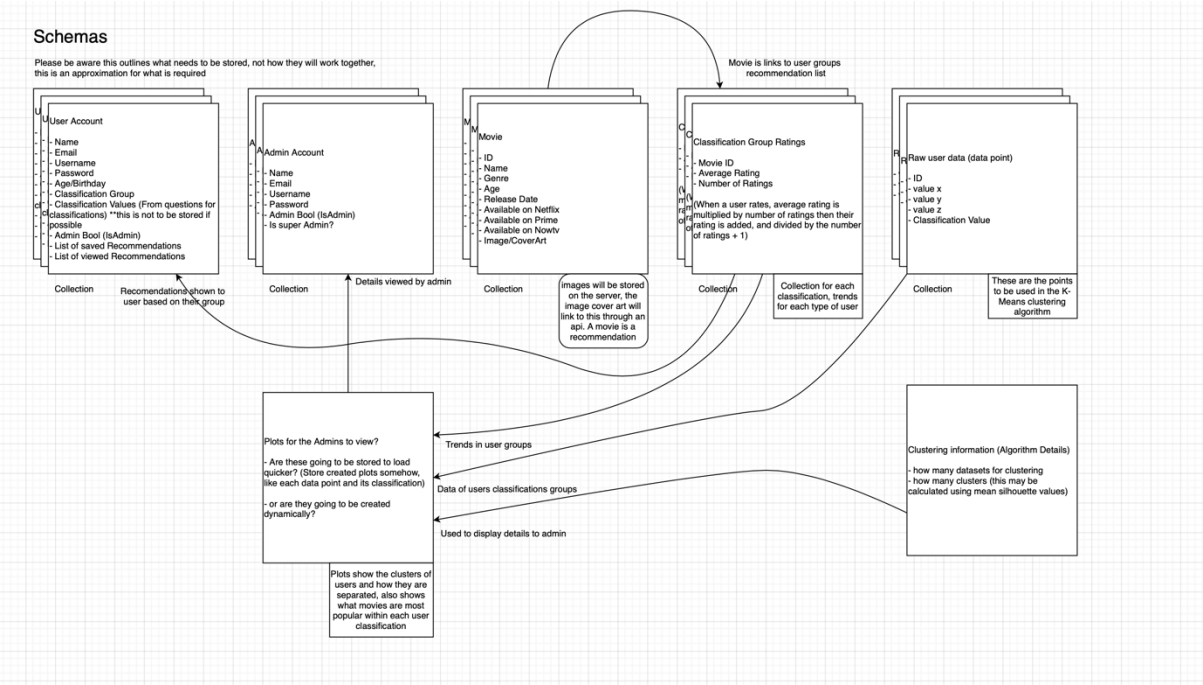
[Figure 8 – Flowchart demonstrating the algorithm for the classification of a user using k-means.]

A great way of visualising user stories is to generate a use case diagram. With a use case diagram, we can see how a user will interface with each function of the software, with each of these functions being based upon a user story or functional requirement. The Use case for ‘Rico Assistant’ (See Figure 9 below), outlines each interaction a user shall make, also each type of intended user for the software is also outlined.



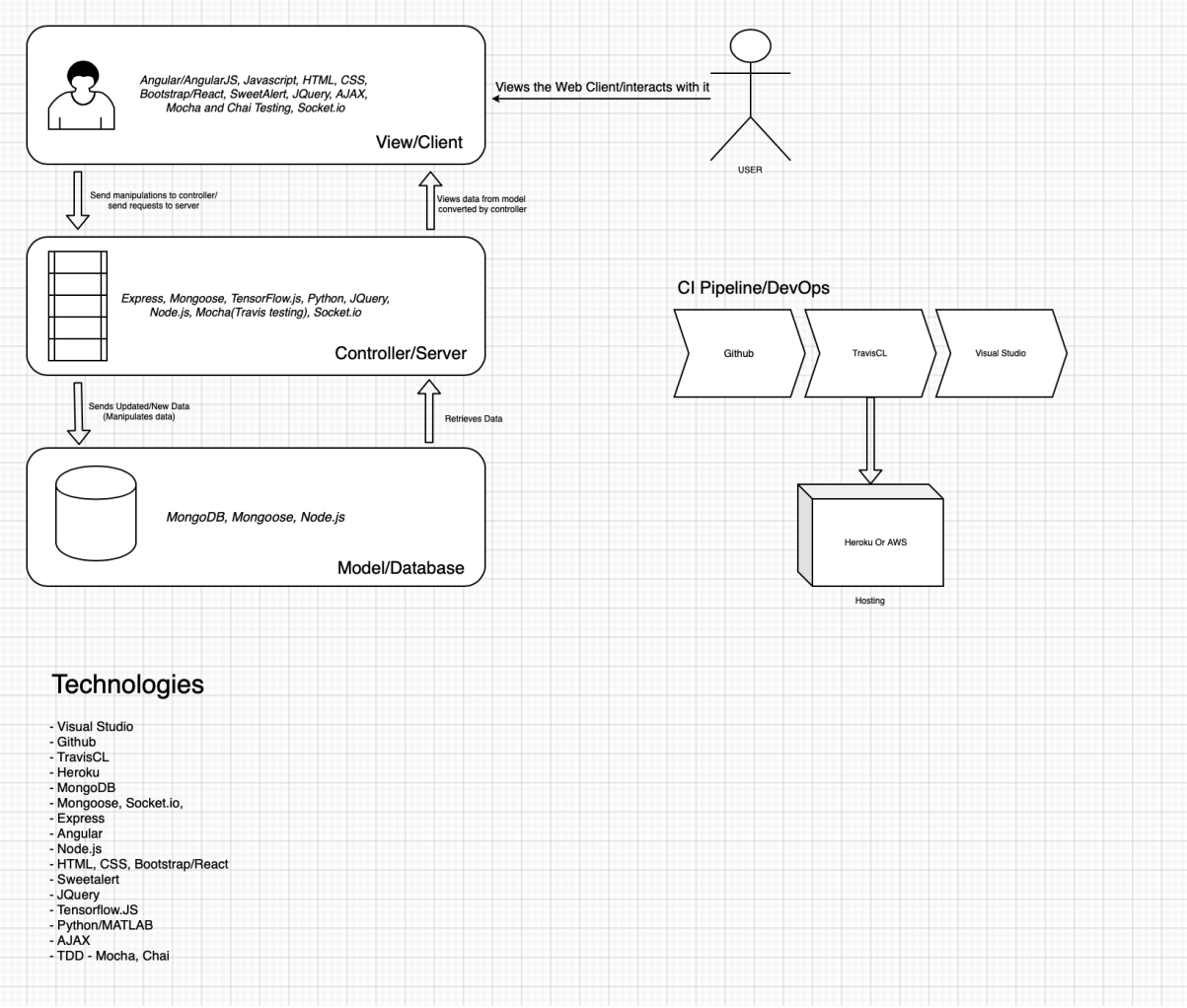
[*Figure 9 – Use Case Diagram for ‘Rico Assistant’*]

With these plans visualised, thought into the structure of the technology stack and database backend can be outlined. The backend database is where everything is stored in a neat and efficient manner. By implementing correct planning, we can ensure that everything that ‘Rico Assistant’ requires to be stored for functionality, is stored in the correct structure. See figure 10 below for the noSQL database structure outline.



[*Figure 10 – Mongo DB Database/Schemas Plan.]*

The technology that ‘Rico Assistant’ relies on is yet another vital aspect that needs to be planned in order to ensure that everything required to run is incorporated. With this, the certainty that what is required can be obtained and more importantly, check technologies are compatible with each other. The intended plan for ‘Rico Assistant’ is to be based on a MEAN Web development stack along with a solid continuous integration pipeline. See figure 11 below for the created plan for the technology stack.



[*Figure 11 – Intended Technology Stack for 'Rico Assistant'*]

5.3.7 User Interface (UI) Designs

It was first thought to design all UI at the start of development in the planning phase, however, it was ultimately decided to create them in an Agile manner. Each UI design (See all in appendix 13.10), was created at the start of a particular page's development as to be certain the UI design would cover all functional and related requirements for the related sprint. Getting a user interface is crucial, however with the responsive and user focused aim, its vital that each and every one is well thought out to achieve this.

The login and sign up pages display the colour scheme of Rico at the fore-front, as a means of differentiating between a logged in and out user, or inside and outside the application. The overall style of the application is aimed at being minimalist as to not give the user too much information that could overpower them, in addition to creating a tidier look and feel. Each page has matching navigation bars that slide out from either side of the application, which gives intuition to the user. The left side is for recommendations and the right is for account and personal pages.

6. Development

An outline of the flow of development is below, in the form of a summary of each sprint week, giving the goal and development overview for each sprint. To see each related Trello board / Product backlog, relating to the progress of these sprints, refer to appendix 13.2. To see the full sprint planning document, refer to appendix 13.3, which includes all sprint reviews and associated information to each sprint.

6.1 Week 1: Project Start-up and Requirements Analysis

Goal

All things that are needed for this project are set up by the end of this week, this includes GitHub, Trello and the vital requirement documents (User Stories, Functional Requirements, Project outline, Basic Gannt Chart).

Development Overview

Here, the basics of the development pipeline were established, the source repository was set up and basic timings were planned. A full functional requirement list (Appendix 13.5) of everything Rico Assistant should accomplish was created this sprint. From the functional requirements, a user story document (appendix 13.8) was also created.

6.2 Week 2: UI Designs and UI Flow-Charts

Goal

This week UI designs, and UI flow charts are to be created. From lasts weeks planning we should be able to see what the UI needs to allow the user to be able to perform. Also, along with the UI designing the logo and branding of the project can be outlined.

Development Overview

As mentioned, UI designs were thought to be created this week to completion, but it was decided here to create them along the Agile style of development when required. The UML flow charts were created and allowed for perfect outlines of how each functional requirement shall be fulfilled along with a more technical visualisation of how things should function. Figure 6 is especially good for demonstrating how a user would progress through the software.

6.3 Week 3: Structure, Technology Requirements and Design UMLS

Goal

The goal for this week is to outline the backend of the software and all its technology requirements. Diagrams should be produced in order to better describe and show intentions of technology plans. If in the case of multiple approaches being

discovered, each approach should have its pros and cons outlined to aid the final decision on approach.

Development Overview
All remaining UML diagrams were created to a good standard, now the whole outline of the project was complete. Possible technology stacks pros and cons were weighed up with the resulting choice being a MEAN stack, based on experience of development and wide use. A bonus to this week was a meeting with Chris Winter, extra and more in-depth planning ideas were given here, and time put aside next week to create them. This week the focus on a movie recommendation system came as after speaking to the project supervisor and Winter, the project scope had to be re-adjusted due to the amount of research and ethical approval (medical applications) to complete such a project.

6.4 Week 4: Risk Assessment, Ethics and Contingency time.

Goal
This week, as all previous plans should have been completed by this point in time, an extensive risk assessment plan can be produced. All risks that could potentially occur should be outlined as well as what precautions should be made in order to minimise their risk, in addition to this a risk action plan should be put into place just in case the risk occurs as to be prepared for the worst. Ethical investigations need to be performed as to ensure that the software complies to legal and ethical regulations set out by law and the university. Finally, the Gannt chart can be updated in order to accommodate each user story, each user story should be awarded points in order to allocate time available each week to each user story/task.

Development Overview
An extensive risk assessment (appendix 13.9) was completed this week as planned, and for the ethical approval no further work was required as advised by the project supervisor. This was due to the Plymouth University ethical policy for the project covering what was required. Documents suggested in the previous weeks sprint with Chris Winter were drafted this week, these were compiled with all other plans created and prepared for a technical advice meeting.

Plans were now considered complete through a meeting with a technical advisor David Walker. In this meeting, a run-down of all the plans was given, this gave confirmation that the plans created thus far were sufficient at outlining the project; through Walkers' advice. Walker also confirmed that the suggested technology stack was the right choice and approach supported by questions raised and answered to defend the projects choice.

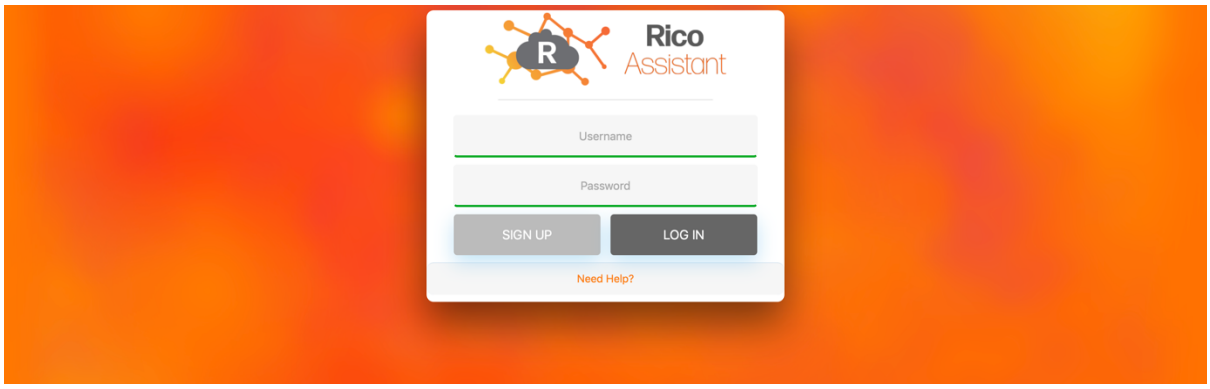
6.5 Week 5: Create Account Page and Login (Start of development).

Goal

User is greeted with a login screen on the loading of the application, a signup page will available in order to create a new account. HCI principles and basic security such as encryption and checking user privileges should be implemented. There should only be one centralised login for users and admins.

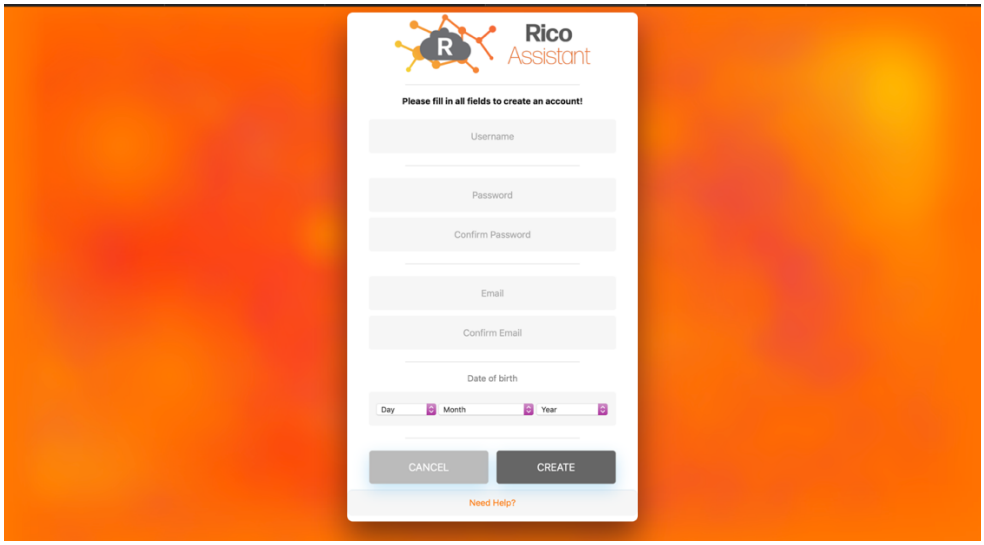
Development Overview

The mark of software development on ‘Rico Assistant’ started here, with slow progress at the start due to under-estimating the workload of a project of this nature. Test driven development slows down fast progress of adding features but will save time in the long run. The final development pipeline was created and tested to confirm that continuous deployment and integration was ready, with all the necessary tests run at each stage.



[*Figure 12 – User Login Page created*]

The login page created (Figure 12) with the client-side testing goes above and beyond what was originally expected with error messages being displayed to the user. This is to follow the HCI design principles strived for in this project. This follows through to the Create account (Figure 13) page, where the error checking really benefits this page. All intended security for this page was implemented as necessary.



[*Figure 13 – Create Account Page*]

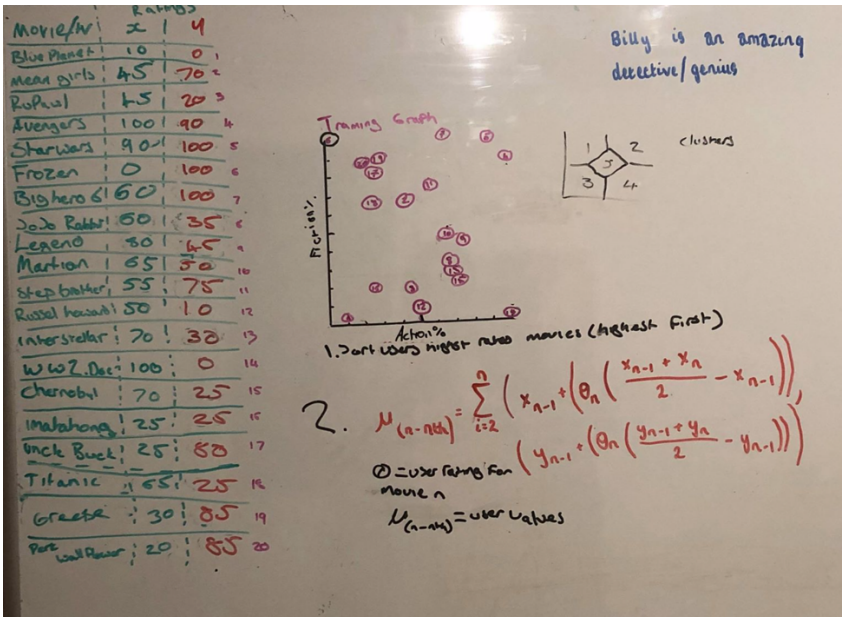
6.6 Week 6: User Classification using K means

Goal

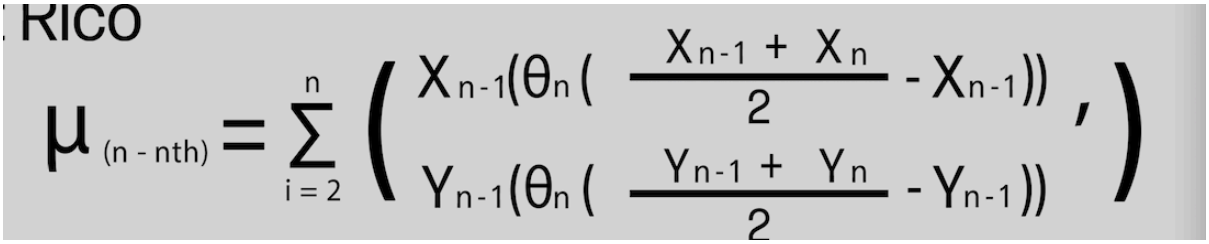
When a user creates an account/on first login/ no classification details are on record, a user is asked questions that will classify them into a cluster/classification group that will allow them to share moving ratings for a group of similar users.

Development Overview

This sprint took much longer than expected as working out a classification scheme that would 'jump start' classification to avoid the cold start issue proved difficult. A whiteboard approach (Figure 14) allowed for a custom algorithm (Figure 15) to be created. This algorithm takes the highest 10 user ratings in order, and then finds a middle ground between them, then returns user classification values ready for k-means.

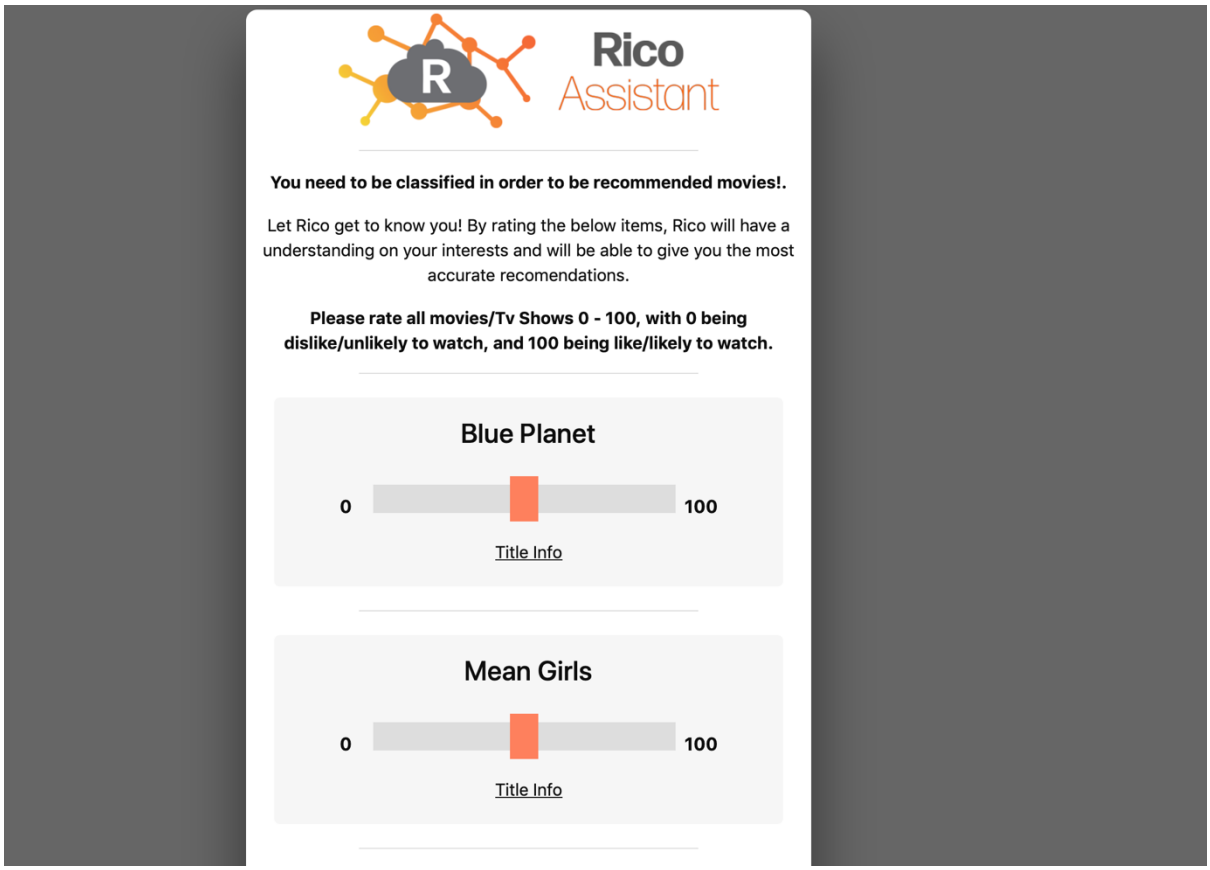


[*Figure 14 – White board prototype of classification algorithm*]



[*Figure 15 – Finished user classification algorithm*]

With a working classification algorithm, a prototype was generated in MATLAB (Figure 5) in order to be certain that a k-means classification of users would generate the desired clusters of users required for the recommendation system.



[*Figure 16 – User Classification Page*]

User classification page was next (Figure 16), and this allowed a user to rate movies in order to jump start their classification in the algorithm resulting in their user account being given a classification group to share recommendations. With this complete a user could now be classified using a custom algorithm and clustered in a group using K-means; the basis of this recommendation system.

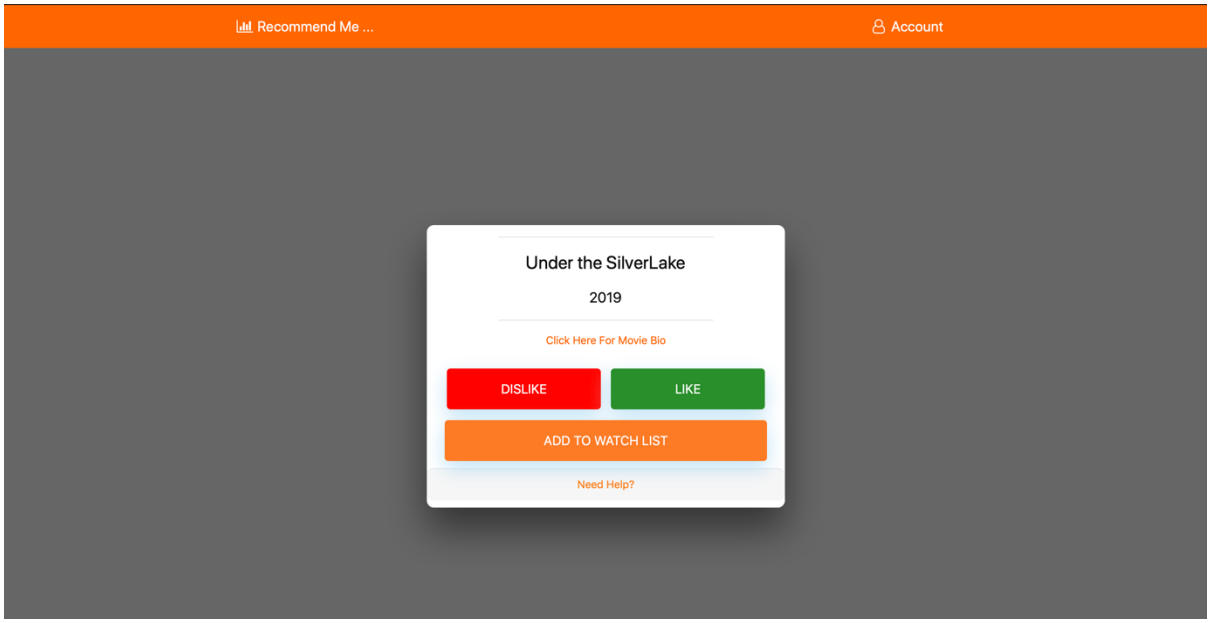
6.7 Week 7-8: User Recommended Related Movies

Goal

Users, when logged in, can go through a visual stack of recommended movies that users of similar statistics (matching cluster group) have rated highly. Best matches/highest rated in that cluster group to be displayed at the top of the stack. User can rate yes or no for the recommendation or say to save to favourites.

Development Overview

Now a user could be classified, a way to see that clusters recommendations had to be designed. From the flow charts of the designed recommendations, the UI was drawn up and implemented to create the movie recommendation page (Figure 17). This was simple enough given the flowcharts already created outlined exactly each function and its intended use. Full functionality was achieved in 3-days for this sprint.



[*Figure 17 – Movie Recommendation Page*]

6.8 Week 9: Users Saved lists and Account Settings

Goal

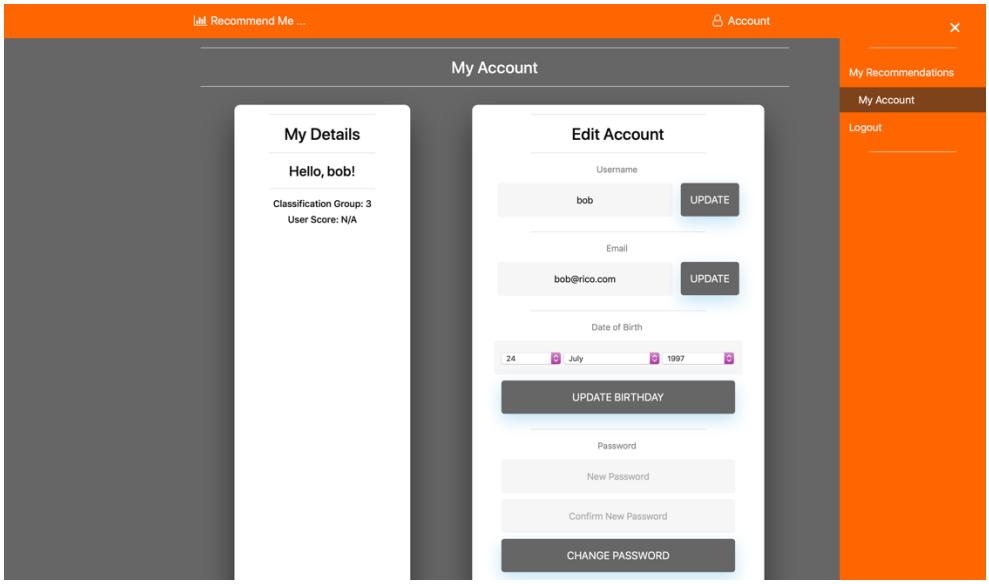
Users can view their account details and stats and change aspects such as password, email, and even delete their account. They can ask to re-classify their group and also if wanted, delete their account. Users also have the ability to view their saved list of recommendations and edit this list, such as reject/delete or return it to circulation of their recommendations.

Development Overview

The account page for the user (Figure 18) due to multiple server functions requiring to be implemented, multiple tests took up several hours of development time. However, all user details can be updated as well as all fields being prefilled as per

good HCI design. User can reset any likes/ratings and also reclassify if they require. As for users editing their lists, this was moved to the users' home page as to be away from the users account details. Note, that even though the location moved, the functionality is the same.

As of this point, a minimum viable product has been reached.



[*Figure 18 – User Account Details Page*]

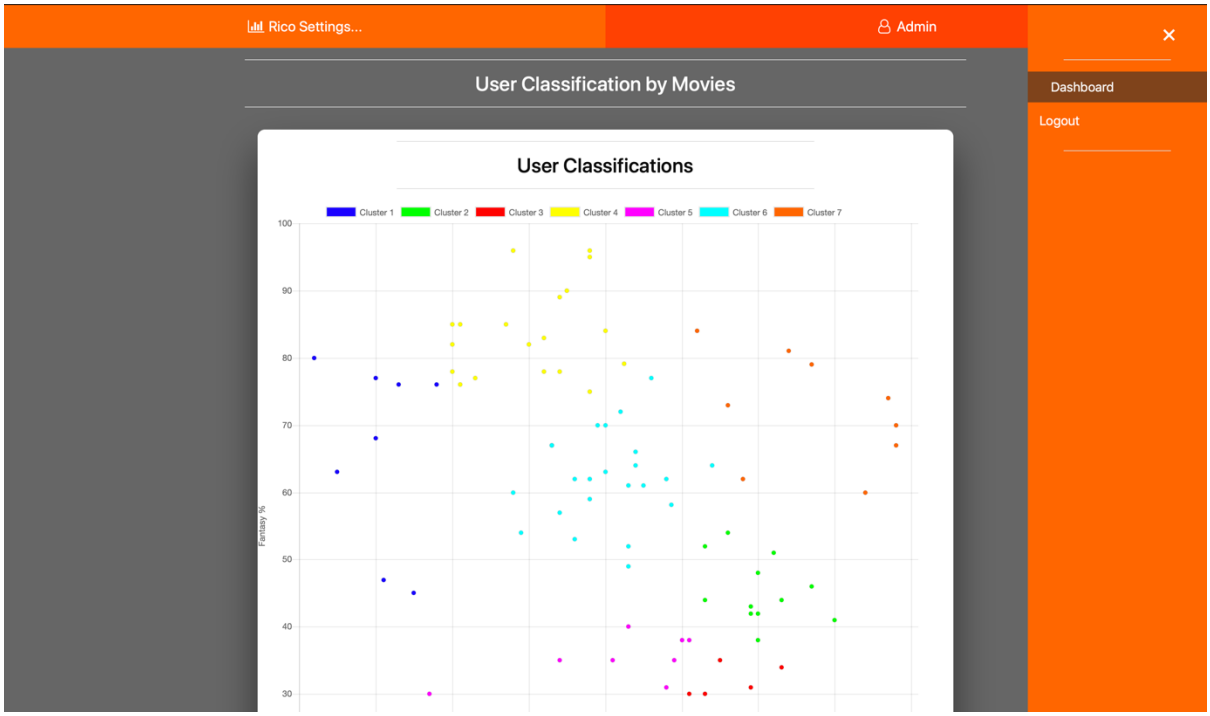
6.9 Week 10: Admin Views

Goal

Admins can view details about the software and how it is operating, that can see plots of user classifications and details about the user base. They can see trends in most popular items that are being recommended and the least.

Development Overview

The views created for the admin with the admin dashboard (Figure 19), although minimal, works very effectively. The Admin can view trends in clusters based on their highest rated recommendations, along with user classification performance from a scatter chart of all users plotted by their classification values. A good addition overall on the MVP.



[*Figure 19 – Admin Views/Dashboard Page*]

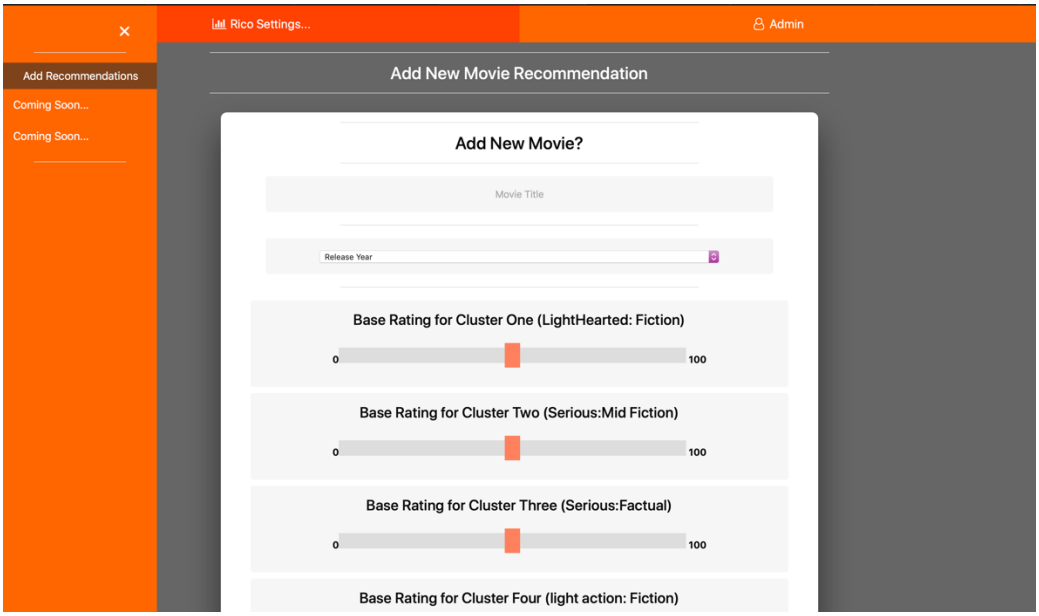
6.10 Week 11: Admin Control of the Algorithm

Goal

Admins are able to edit the algorithm slightly, in order to allow users higher accuracy in the operation of the software. they can also add new recommendations to be added to the users of the software.

Development Overview

Although not everything this week was implemented, an admin can enter new movie recommendations for users to get recommended (figure 20). However, unfortunately admins are unable to adjust algorithm details nor are they able to deal with user classifications. On the other hand, all core and desirable functionality is completed even with some additional functionality. This is the end of development for the application as for adding new features.



[*Figure 20 – Admin Add Recommendations Page*]

6.11 Week 12: User Testing & Final Fixes/Additions

Goal

User testing needs to run in order to test the product with the end user. Through testing we can see if user stories are met and the actual intended user is happy with the product. Evaluation of the findings will tell us of any additions/fixes that need to occur and as these should only be minor, we can fix these in this week.

Development Overview

User testing couldn't have gone better, due to the care and attention that came with focusing Rico's design around UX and UI with focus to the user, as well as being fully responsive, user feedback was in general very positive.

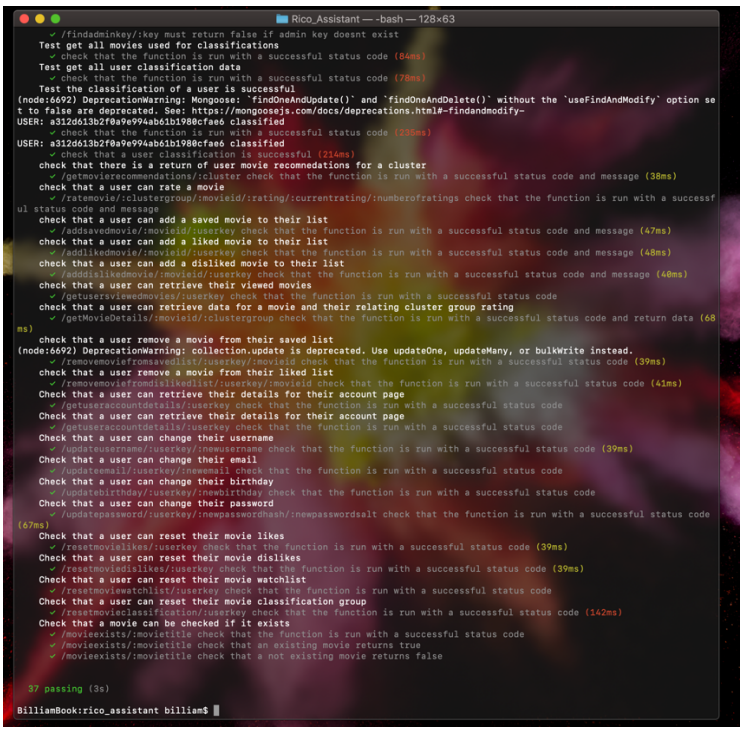
7. Testing

Testing of any software package that is released to the end user is important, it highlights any errors that are in the solution that require to be fixed immediately. By thoroughly testing the software we can deliver a product that is stable, usable, and one that the end user will be satisfied with.

7.1 Unit Testing

Unit testing is the testing of small pieces of code to check whether they are functioning as intended. The benefits of doing so is Agile checking of code, improving the quality of code, finding bugs early, and to facilitate changes through development [*Novoseltseva, 2018*]. As mentioned, this project was developed using test driven development, and so every function of the code is tested with a unit test. There is additional testing in the form of Mocha and Chai on HTML pages at to confirm all elements required to function as an app are present. Furthermore, all tests are run in the pipeline as to ensure with every change/addition of code, no errors to functions are created. Integration Testing is also partly implemented on certain functions by testing multiple units as a whole, this is seen on the testing of some functions such as the node server get requests.

Unit tests were run multiple times throughout development to ensure no ill effects of further development presented themselves. As previously mentioned, this was achieved through Travis.CI and its automated testing, as well as manually testing through Node Package Manager (NPM) periodically (Figure 21).



[*Figure 21 – Manual run of server-side unit tests, ran using the 'npm test' command*]

7.2 Functional Testing

Functional Testing covers all functionality of the software. It ensures that all functionality of the code is giving the results as expected and it delivers a level of quality assurance to the released product. By following this procedure, a defect-free product can be delivered, it ensures the end user is satisfied, and ensures that all requirements are met [XenonStack, 2018]. Being based around functionality, a functional test plan (Appendix 13.11) was created to test each functional requirement within the code and to see if the end user requirements were indeed met. This test plan was run after every sprint, to test if new additions were functional, or if they affected previous.

7.3 User Testing

To get an opinion on the usability of the software, user acceptance testing was performed in order to get a direct response from users of using ‘Rico Assistant. Analysis into user’s response to the software can give an insight into aspects such as user satisfaction and problem areas, if multiple test users experience similar problems, this is an indication of an aspect of improvement. Most of all, User testing allows for an unbiased review of the project [*Williams, 2019*].

A user test group was comprised of final year computer science students. Each one completed the same set of tasks and completed a survey for both the general user side of the application and the admin side; guidance was given where necessary. The survey and results from this testing can be seen in appendix 13.2.

8. End of Project Report

8.1 End of Project Summary

In all, the project proved to be a success. The whole premise of the project was to develop a recommendation system that is simple and easy to understand, all while tending to some of the issues of existing recommendation systems. This goal was reached as proved by the several testing methods, as a working recommendation system that the end users approve of, was attained. The secondary goal of being user focused is also a monumental success due to the HCI design and careful attention to UX and UI design. The application functionality doesn't disappoint either, with all core and desirable goals attained, with a good few additional requirements also sprinkled in.

8.2 Planned Design, Objectives & The Finished Product

When comparing the finished product to the planned designs, the project's success is evaluated. The original plans dreamed of a project of a much higher scope; however, this was known from the start as the scope was designed have more than enough development, to be a project with 'growing room'. With the design of the project being so strong, at no point was there ever doubts on what to implement, with the strongest idea of what Rico has to achieve being from all UML flow charts.

The main goals set to achieve from the project vision and ideas were to:

1. Create a simple, yet effective recommendation system
2. Get past limitations in current recommendation systems
3. Ensure solid HCI and UX design with responsive UI
4. Accessible application to all users
5. To recommend movies to a user and be accurate in suggestions.

Starting with 1, a simple recommendation system using k-means was developed, Rico's simple to understand but more importantly, as proven by user testing is effective at supplying the goal to the end user. This continues with 2, as the three downfalls of collaborative recommendation systems are mostly tended to within Rico. There are no known cold start issues or sparsity issues, however as for scalability further research is required. Rico Assistant is prepared for scalability within plans shown in the classification algorithm flow-chart (Figure 8), with a means of splitting the datasets of users up into more digestible parts.

3 is fully met, at no point does Rico Assistant fall from a solid approach in regard to HCI, UI and UX design. The whole project revolved around ensuring at every stage user satisfaction should be high, this is discussed briefly in the next section. Being an app of a responsive nature, the application is fully accessible to all users satisfying goal 4. Finally, goal 5, the application does recommend movies to users that users are of interest, shown in the general user testing results question 5 (Appendix 13.12), with a very positive result.

Changes from plans were present, although this is expected of a larger scale software development project. Arguably the finished product is of a simpler nature, take the original database (Figure 10) that is far over engineered for the required goals of the application. The simpler database (Figure 22) is much more effective and efficient of storing data. All changes from plans are justified, as due to the agile nature of development and the benefits of DRY, if a simpler path was present, it was taken.

```
//table structure for a users account
var User = mongoose.model("Account", {
    Username: String, //Must be unique, checked client side atm
    Password: String, //Salted and Hashed
    Password_Salt: String, // matching salt
    Email: String,
    Birthday: Date,
    UserKey: String, //randomally generated key
    Classification_Group: Number,
    Class_Value_One: Number,
    Class_Value_Two: Number,
    Class_Value_Three: Number,
    Saved_Recommendations: [Number],
    Liked_Recommendations: [Number],
    Disliked_Recommendations: [Number]
});

//table structure for list of approved user keys that are admins
var AdminKey = mongoose.model("auser", {
    Key: String
});

//table structure for a Classification
var ClassificationMovie = mongoose.model("classificationmovie", {
    Name: String,
    X: Number,
    Y: Number,
    Id: Number
});

//table structure for a movie
var Movie = mongoose.model("movie", {
    Id: Number,
    Title: String,
    ReleaseYear: Number,
    Cluster_Zero_Rating: Number,
    Cluster_Zero_Number_Of_Ratings: Number,
    Cluster_One_Rating: Number,
    Cluster_One_Number_Of_Ratings: Number,
    Cluster_Two_Rating: Number,
    Cluster_Two_Number_Of_Ratings: Number,
    Cluster_Three_Rating: Number,
    Cluster_Three_Number_Of_Ratings: Number,
    Cluster_Four_Rating: Number,
    Cluster_Four_Number_Of_Ratings: Number,
    Cluster_Five_Rating: Number,
    Cluster_Five_Number_Of_Ratings: Number,
    Cluster_Six_Rating: Number,
    Cluster_Six_Number_Of_Ratings: Number
});
```

[*Figure 22 – Simplified final implementation of the database structure*]

8.3 Effectiveness with the End User

Satisfaction from the end user, overall was very positive in the user acceptance testing. This was all achieved as from the very start, care and due attention to the end user was applied. This is demonstrated from the plans created through to the finished product, with documents such as functional requirements and user stories being the driving focus of implementations into the software package. No sprint week was worked on without relating all progress to the end users' needs.

9. Project Post-Mortem

9.1 Method of Approach & Development Review.

Approaching the project with an Agile mindset allowed the project to never fall too far from schedule, in addition to bringing all the benefits mentioned with an Agile development cycle. Unfortunately, due to the developer not implementing a recommendation system before, it was not fully known how long some of the algorithm creations take. This set back and disturbed the schedule of sprints for some time, with many sprints overlapping and not sticking directly to schedule. On the other hand, with Agile accepting and accounting for flexibility, a solution of re-adjusting was easy, this confirms that agile was the right approach for development.

9.2 Was the End Project Effective at Satisfying Set Out Objectives?

Rico Assistant was more than effective at reaching the set-out objectives of the project. However, the projects development and evaluation could have benefited for clearer identifiers of each objective, a set document outlining each objective should have been created in hindsight. Too much development time was spent on planning, this was due to worry of not fully covering all of Ricos design. Alternatively plans could have been broader, with each agile sprint week focusing on the required aspects. Overall Rico fulfils the goals set out and attends to the problems of recommendation systems in function today.

9.3 Best Aspects

Notable aspects that should be taken away are, the custom algorithm for classification as this is the key to creating a simpler recommendation system. It's amazing how such a simple implementation of a recommendation system is fully functional and arguably better, along with a more user focused design. The user focus, along with the fantastic UI/UX and HCI design greatly improves user satisfaction and delivers an effective solution to the end user. Finally, being supported by a robust, sturdy and invaluable development pipeline allows for constant update delivery to the end user, with every release automatically tested to ensure quality.

9.4 Future Work

Rico was always designed to have future work, with the main goal of development to set up a framework recommendation system that can recommend items from a selection of areas. So, more recommendations are to be added in future release utilizing the fundamentals of the movie recommendation system created.

In plans, more specifically the use case diagram (Figure 9), a super admin could benefit the project. This super admin could approve changes from lower admins as to keep to misuse stories of admins breaking the software as well as adjusting the algorithm through the software, as opposed to developing them in. This was mentioned as an additional requirement and these would be completed in future work. Another mentioned addition, a neural network replacement for k-means, could be implemented for further accuracy of recommendations, however this could increase the complexity of the application, decreasing the commended simplicity of Rico Assistant recommendation system.

9.5 Developer Performance Review

Summarising the developer's performance, on the most part was more than acceptable. The planned 30 hours of development a week was reached, given the time available the amount of effort applied is more than acceptable. Although the project may not have the dreamt for functionality, a professional product achieved doesn't have much else that could have been implemented. Any short comings and time delays come down to the developer's time taken to research new ideas and coding practices in order to continue development, as some aspects the developer had not experienced before. The challenge of the unknown territory kept the project interesting, assisting in the momentum of development. It's truly believed that given all circumstances and time available, the project and development could not have

10. Final Words / Conclusion

Concluding, the project of Rico Assistant achieved mostly what it was set out for. I'm very proud of the application achieved along with the documentation that was produced. I feel it's a good representation of the high standard I strive for even if it could be argued that too much attention to detail slows development. Although the terms of this project the application is complete, the application has much more work available, with later features and research to conclude. When times were stressful towards the end of development and time limitations ensued, it really improved creativity and reuse of code, with multiple functions being re-used where possible.

11. References

Atlassian. 2020. *What Is Agile?* | *Atlassian*. [online] Available at: <https://www.atlassian.com/agile> [Accessed 25 April 2020].

Baghel, A., 2018. *Software Design Principles DRY And KISS - Dzone Java*. [online] dzone.com. Available at: <https://dzone.com/articles/software-design-principles-dry-and-kiss> [Accessed 27 April 2020].

Boam, E., 2019. *I Decoded The Spotify Recommendation Algorithm. Here's What I Found.*. [online] Medium. Available at: <https://medium.com/@ericboam/i-decoded-the-spotify-recommendation-algorithm-heres-what-i-found-4b0f3654035b> [Accessed 22 April 2020].

Bobadilla, J., Ortega, F., Hernando, A. and Bernal, J., 2012. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26, pp.225-238.

Breese, J., Heckerman, D. and Kadie, C., 2013. *Empirical Analysis Of Predictive Algorithms For Collaborative Filtering*. [Paper] Redmond.

Cohn, M. (2009). *User stories applied*. Pearson.

Deng, H., 2019. *Recommender Systems In Practice*. [online] Medium. Available at: <https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a> [Accessed 22 April 2020].

DPO Centre. 2018. *What Is The Difference Between The DPA 2018 And The GDPR?*. [online] Available at: <https://www.dpocentre.com/difference-dpa-2018-and-gdpr/> [Accessed 24 April 2020].

Git-scm.com. 2020. *Git - About Version Control.* [online] Available at: https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control [Accessed 26 Apr. 2020].

Git Tower, 2020. *Why Use Version Control?*. [online] Git-tower.com. Available at: <https://www.git-tower.com/learn/git/ebook/en/command-line/basics/why-use-version-control> [Accessed 25 April 2020].

HASELT. 2016. *Coding Dojo with TDD.* [online] Available at: http://haselt.com/coding-dojo-with-tdd/ [Accessed 25 Apr. 2020].

Hardesty, L., 2019. *The History Of Amazon's Recommendation Algorithm*. [online] Amazon Science. Available at: <https://www.amazon.science/the-history-of-amazons-recommendation-algorithm> [Accessed 23 April 2020].

Help Center. 2020. *How Netflix's Recommendations System Works*. [online] Available at: <https://help.netflix.com/en/node/100639> [Accessed 22 April 2020].

Heroku, 2020. *20120524 Policy | Heroku*. [online] Available at: <https://www.heroku.com/policy/20120524-policy> [Accessed 23 April 2020].

Hosein Jafarkarimi, Alex Tze Hiang Sim, and Robab Saadatdoost, A Naïve Recommendation Model for Large Databases, *International Journal of Information and Education Technology* vol. 2, no. 3, pp. 216-219, 2012.

ICO: Information Commissioners Office, 2018. *Guide To The General Data Protection Regulation (GDPR)*. UK government. Available at: <assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/711097/guide-to-the-general-data-protection-regulation-gdpr-1-0.pdf> [Accessed 23 April 2020].

Katalon, 2019. *Top 10 Benefits Of Continuous Integration & Continuous Delivery*. [online] Available at: <https://www.katalon.com/resources-center/blog/benefits-continuous-integration-delivery/> [Accessed 25 April 2020].

Legislation.gov.uk, 2018. *Data Protection Act 2018* [online]. Available at: http://www.legislation.gov.uk/ukpga/1999/28/contents (Accessed: 30 April 2020).

Littlefield, A., 2016. *The Beginner's Guide To Scrum And Agile Project Management*. [online] Blog.trello.com. Available at: <https://blog.trello.com/beginners-guide-scrum-and-agile-project-management> [Accessed 25 April 2020].

Lü, L., Medo, M., Yeung, C., Zhang, Y., Zhang, Z. and Zhou, T., 2012. Recommender systems. *Physics Reports*, 519(1), pp.1-49.

Moran, A. 2016. Risk Management in Agile Projects. *ISACA*, [online] 2. Available at: https://www.isaca.org/resources/isaca-journal/issues/2016/volume-2/risk-management-in-agile-projects [Accessed 23 Feb. 2020].

Moreno, M., Segrera, S., López, V., Muñoz, M. and Sánchez, Á., 2016. Web mining based framework for solving usual problems in recommender systems. A case study for movies′ recommendation. *Neurocomputing*, 176, pp.72-80.

Novoseltseva, E., 2018. *Benefits Of Unit Testing, Our List Of The Top 8 | Apiumhub*. [online] Apiumhub. Available at: <https://apiumhub.com/tech-blog-barcelona/top-benefits-of-unit-testing/> [Accessed 27 April 2020].

Opensource.org. 2020. *The MIT License | Open Source Initiative*. [online] Available at: <https://opensource.org/licenses/MIT> [Accessed 23 April 2020].

Opensource.com. 2020. *What Is Open Source?*. [online] Available at: <https://opensource.com/resources/what-open-source> [Accessed 22 April 2020].

Polatidis, N. and Georgiadis, C., 2013. Recommender Systems. *International Journal of E-Entrepreneurship and Innovation*, 4(4), pp.32-46.

Sami, M., 2012. *Software Development Life Cycle Models And Methodologies - Mohamed Sami*. [online] Mohamed Sami. Available at:

<https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/> [Accessed 25 April 2020].

Smashing Magazine. 2011. *Responsive Web Design - What It Is And How To Use It — Smashing Magazine*. [online] Available at: <https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/> [Accessed 27 April 2020].

Son, B., 2019. *A Beginner's Guide To Building Devops Pipelines With Open Source Tools*. [online] Opensource.com. Available at: <https://opensource.com/article/19/4/devops-pipeline> [Accessed 25 April 2020].

Stemeir, S., 2018. *The Murky Ethics Of Data Gathering In A Post-Cambridge Analytica World*. [online] American Marketing Association. Available at: <https://www.ama.org/marketing-news/the-murky-ethics-of-data-gathering-in-a-post-cambridge-analytica-world/> [Accessed 24 April 2020].

Trello. 2017. *What is Trello? - Trello Help.* [online] Available at: https://help.trello.com/article/708-what-is-trello [Accessed 25 Apr. 2020].

Williams, M., 2019. *The Benefits Of Usability Testing Explained - Keycdn*. [online] KeyCDN. Available at: <https://www.keycdn.com/blog/usability-testing> [Accessed 27 April 2020].

Wong, E. (2018). *Shneiderman’s Eight Golden Rules Will Help You Design Better Interfaces*. [online] The Interaction Design Foundation. Available at: https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces [Accessed 8 April. 2020].

XenonStack. 2018. *Functional Testing Advantages, Architecture And Tools - Xenonstack*. [online] Available at: <https://www.xenonstack.com/insights/what-is-functional-testing/> [Accessed 27 April 2020].

Zhao, Z., Sheng, Y., Zhu, M. and Wang, J., 2018. A Memory-Efficient Approach to the Scalability of Recommender System With Hit Improvement. *IEEE Access*, 6, pp.67070-67081.

12. Bibliography

Cichocki, A., 2009. *Nonnegative Matrix And Tensor Factorizations*. Chichester: John Wiley.

Jannach, D., 2012. *Recommender Systems*. Cambridge: Cambridge University Press.

Littlefield, A., 2016. *The Beginner's Guide To Scrum And Agile Project Management*. [online] Blog.trello.com. Available at: <https://blog.trello.com/beginners-guide-scrum-and-agile-project-management> [Accessed 25 April 2020].

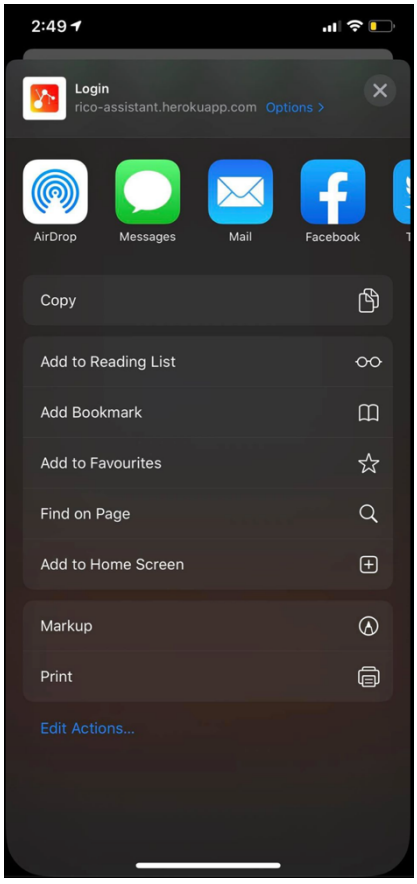
Roman, V., 2019. *Unsupervised Classification Project: Building A Movie Recommender With Clustering Analysis And…*. [online] Medium. Available at: <https://towardsdatascience.com/unsupervised-classification-project-building-a-movie-recommender-with-clustering-analysis-and-4bab0738efe6> [Accessed 28 April 2020].

13. Appendices

13.1 User Guide

Installation

Rico Assistant is a web application hosted at: https://rico-assistant.herokuapp.com
Due to being a responsive application either navigate to the above address on either a mobile or desktop device with your preferred internet browser. On Apple devices Rico can be saved as an app icon by saving a shortcut to within safari and selecting add to home screen. Below is a result of doing so.

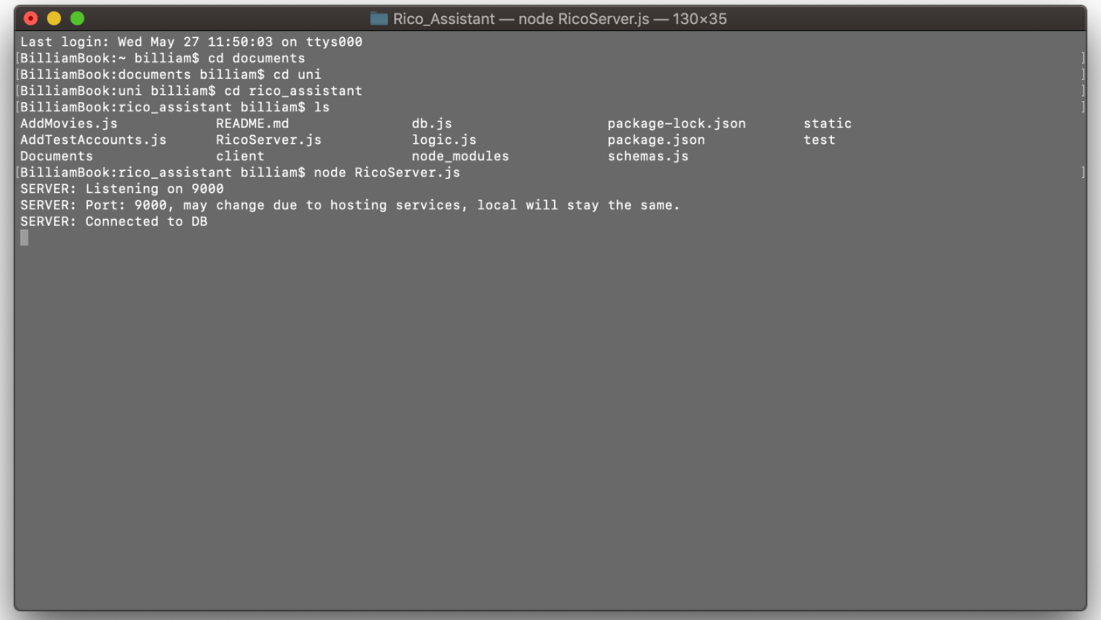


Local Hosting (Development View)

To run locally with visibility of mocha and chai testing results display, as being a node.js application, a locally hosted version can be run and accessed by navigating to ‘localhost:9000’.

In order to run a node application be sure to have node.js installed on your desktop and open a terminal. Navigate to the location of the downloaded code files and open the directory containing ‘RicoServer.js’

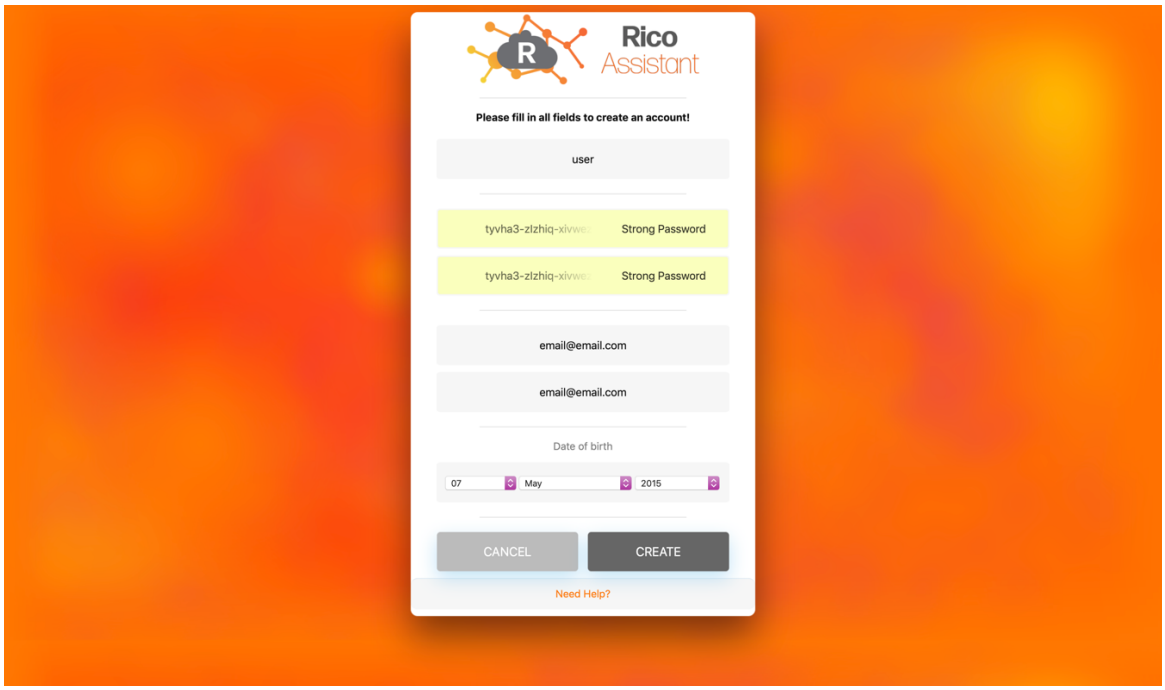
To run to local version run the command ‘node RicoServer.js’



GENERAL USER FUNCTIONS:

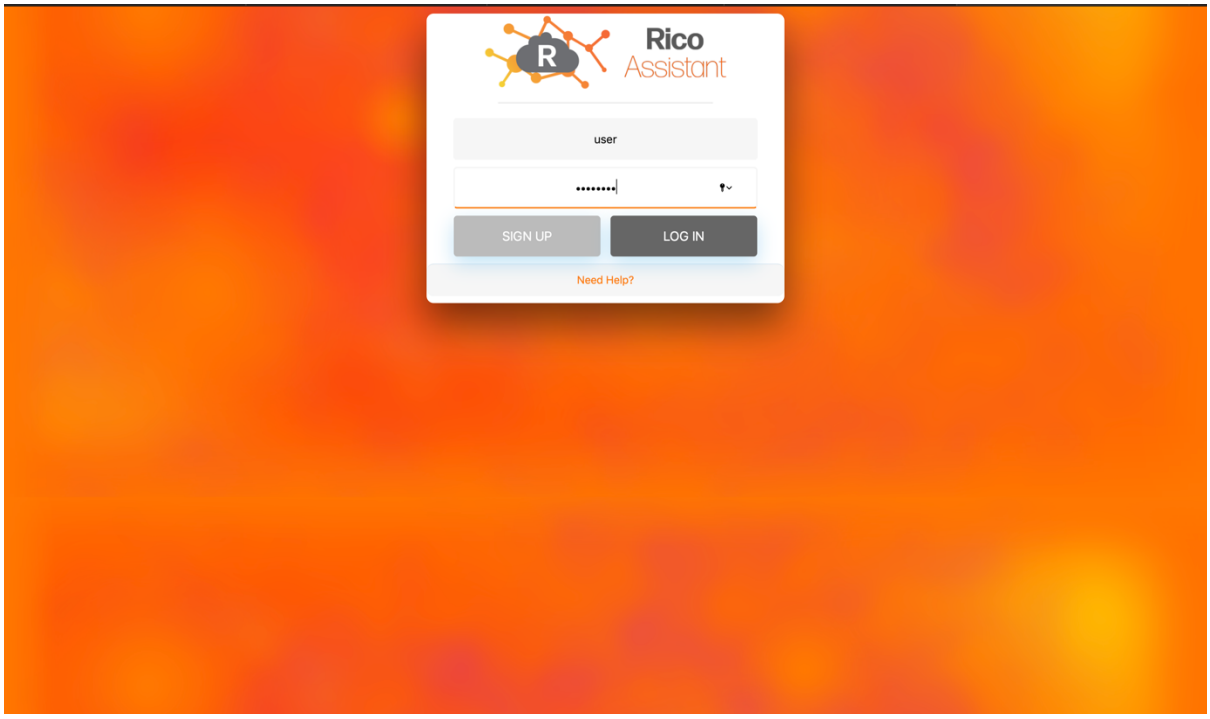
Create Account

To create an account, navigate to the application and click on sign up, this will navigate you to the create account page. Fill in all the fields and remember the details. Click on ‘sign up’ and your account shall be created.



Login

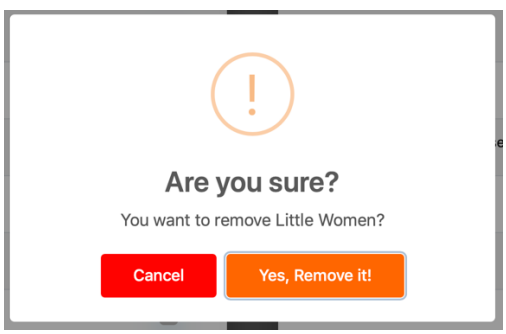
Once you have a created account, to login navigate to the login (greeting page of the app) and enter the login details of your account (username and password). Click login and you shall be taken to the user homepage on a successful login.

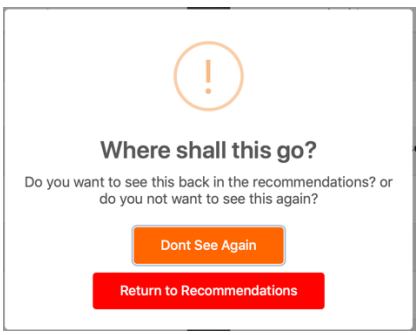


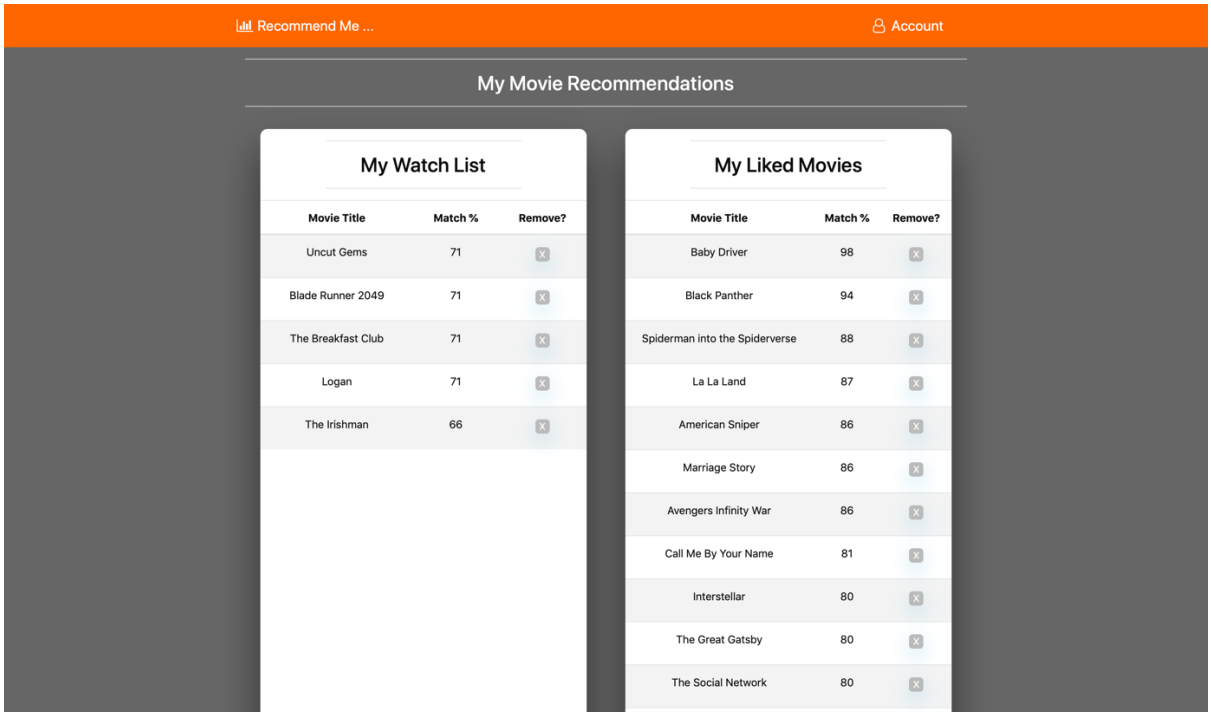
User Homepage

Once logged in you will be greeted with you user homepage, to navigate around the site use the navigation par at the top of the page. The left ‘recommend me..’ button will take you to areas of recommendations for items to be suggested to you. The right ‘account. Button will display navigation options for account related pages.

The page contains details of your likes and watchlist, to remove items these lists click on the ‘x’ for the related recommendation. On removal you will be asked where you would like the removed recommendations to go, either back to recommendations to be suggested or if you choose to not display again it shall be moved to your dislikes.

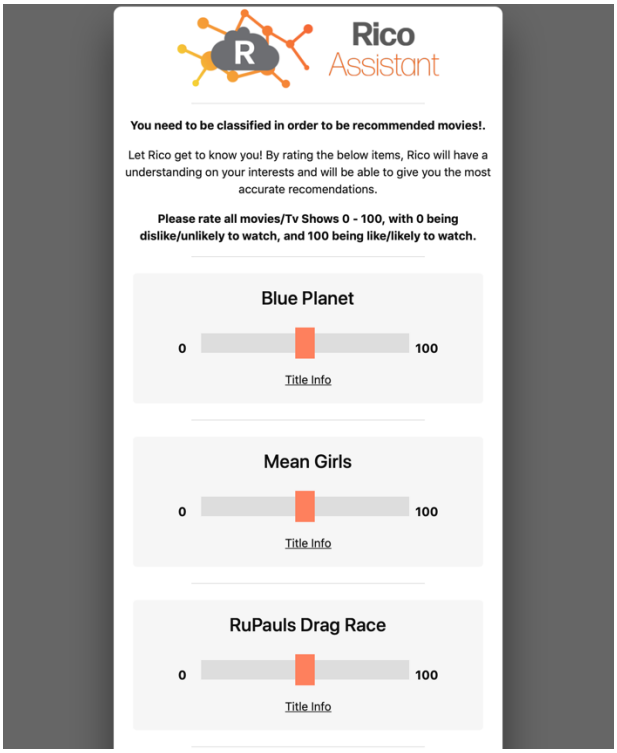






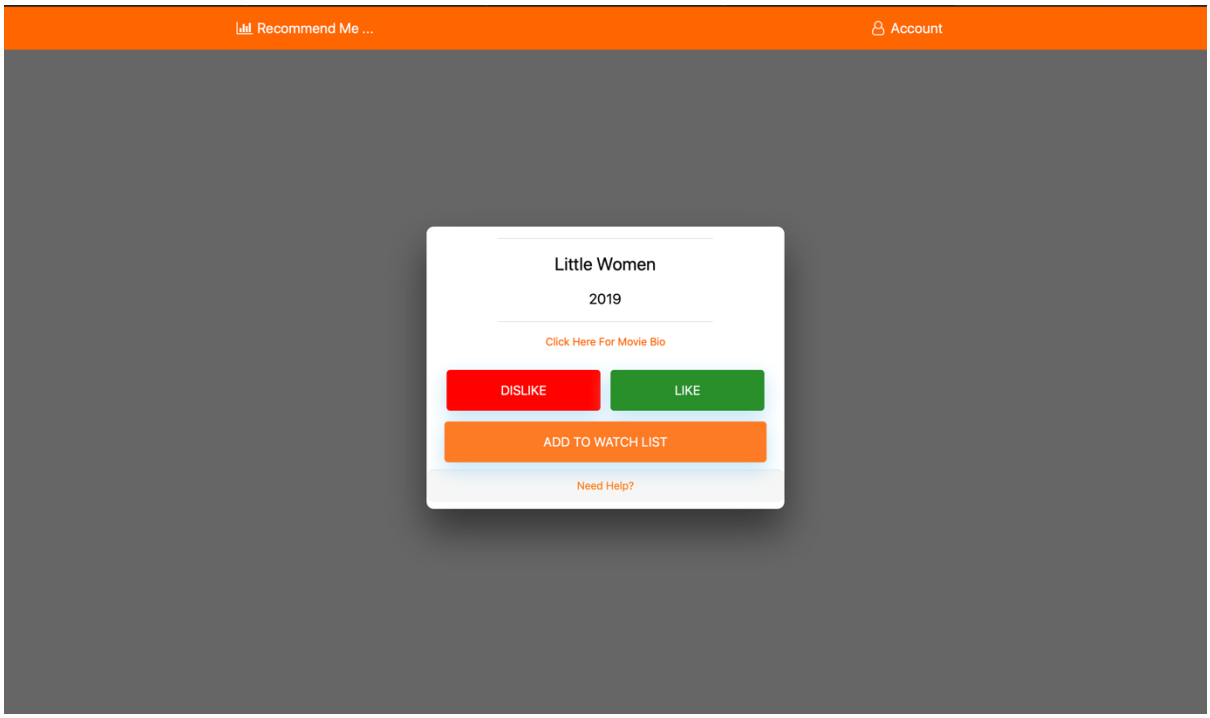
User Classification

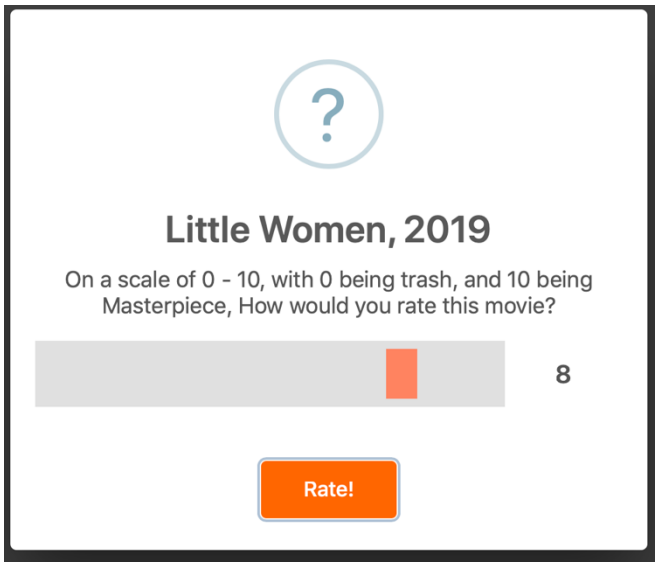
On navigation to a recommendation area you have not been to before, you will be asked to answer questions in order to classify you into a group in order to receive recommendations for the recommendation area selected. Follow the instructions on screen and use the sliders on the page to rate each item in the list. When complete click on the ‘classify me button’



Movie Recommendation

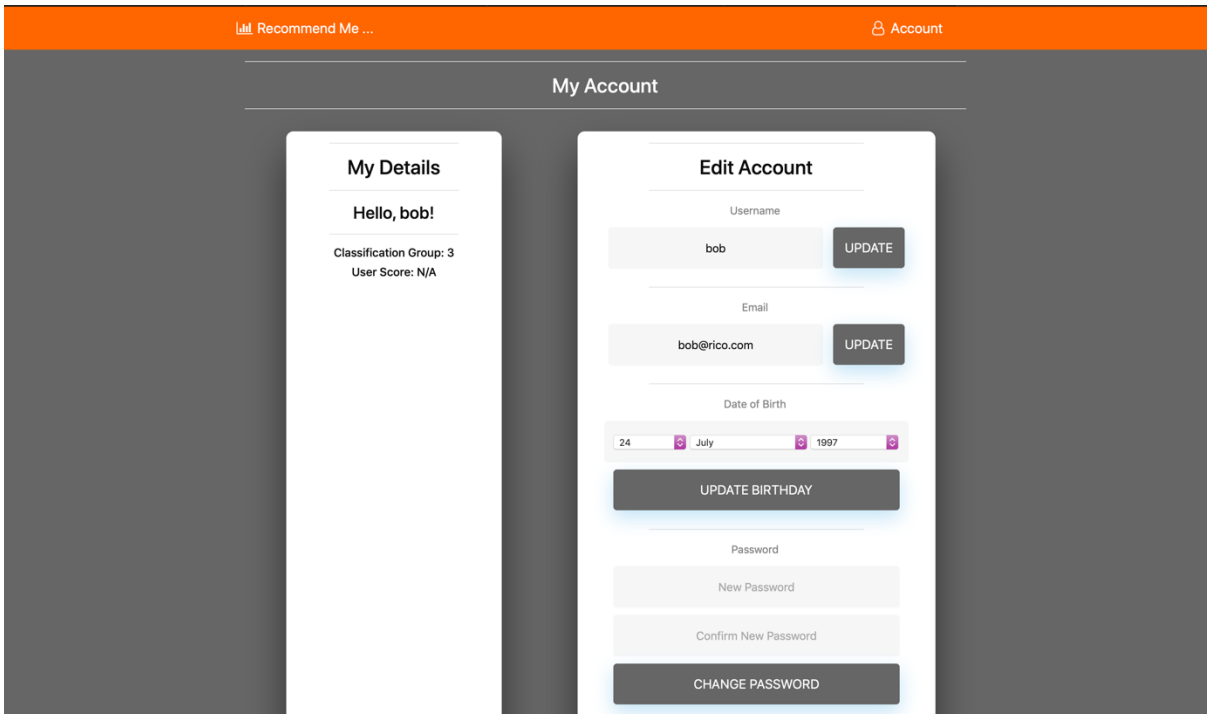
If classified for that recommendation area, the best suited recommendations to you will be displayed with the best first. To view more information, click on the link to view movie bio, this will open a new tab containing information on the movie being recommended. If you like the recommendation click on the like button and you will be asked to rate the movie on a scale of 1-10. To save to your watchlist click the related button and if this was a recommendation that doesn't suit you, click on the dislike button to not display this recommendation again.





Account Page (My Details)

To edit account details, navigate to my details. Here you can edit all your account information by changing the selected field and clicking update. To reset any list click on the corresponding button. Same goes for if you want to be reclassified for a certain area. To delete your account, click the button at the bottom of the page. Be aware this will clear all information associated with your account forever.



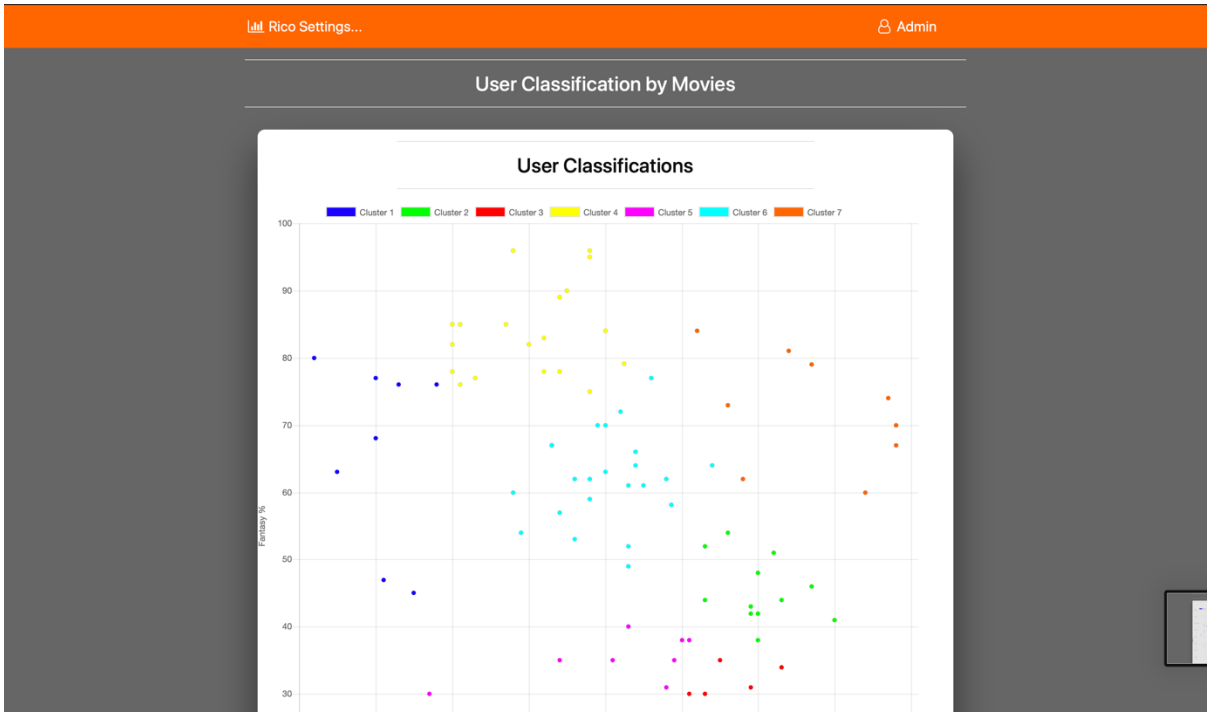
ADMIN USER FUNCTIONS:

Login

To login use the admin username and password (admin, password) and click on login. Rico will know that this account is an admin and take you to the admin dashboard.

Admin Dashboard

When logged in, you will be greeted with the admin dashboard. At the top of the page is the navigation bar. On the left ‘rico settings’, this will bring up a menu of all possible settings you can edit. The right ‘admin’ will display all views and account settings for an admin.

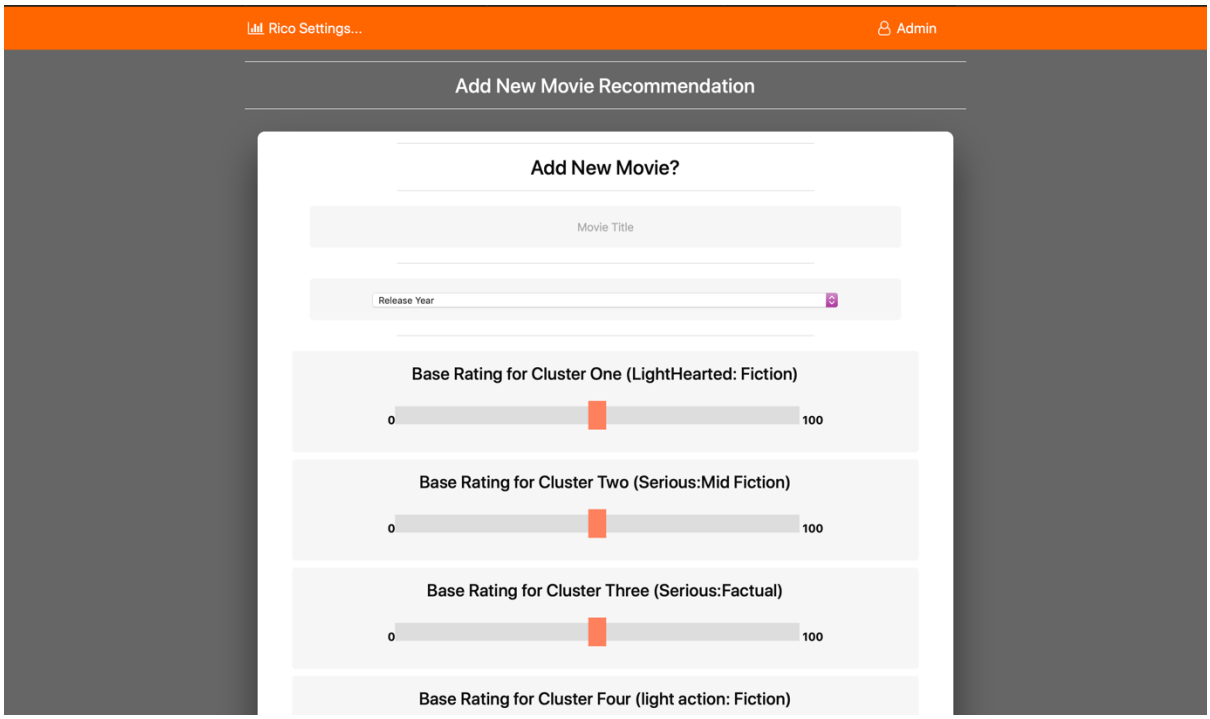


The ‘User classifications’ graph shows all users and their corresponding classification groups plotted with their classification values. Here you can see the current performance of the classification algorithm. You can click on the key to hide/display data for each cluster group.

Lower down the page you will find the trends of recommendations for each cluster group.

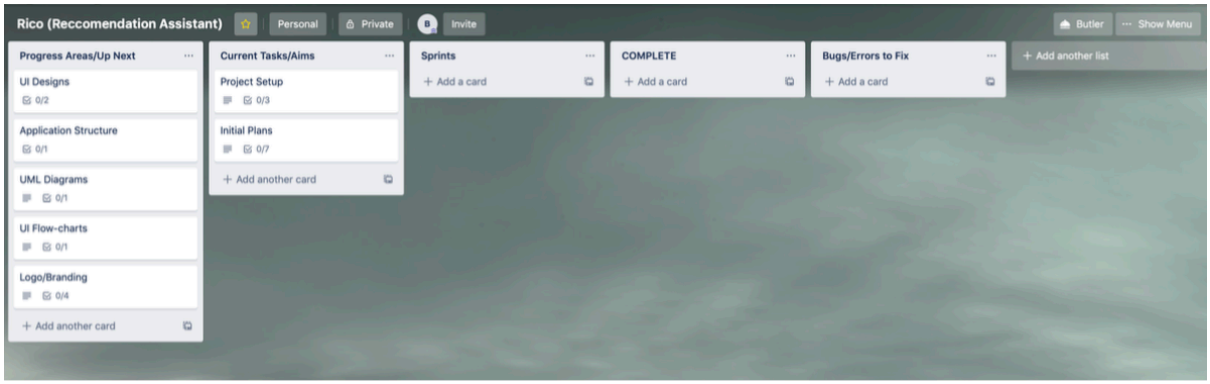
Add Movie Recommendation

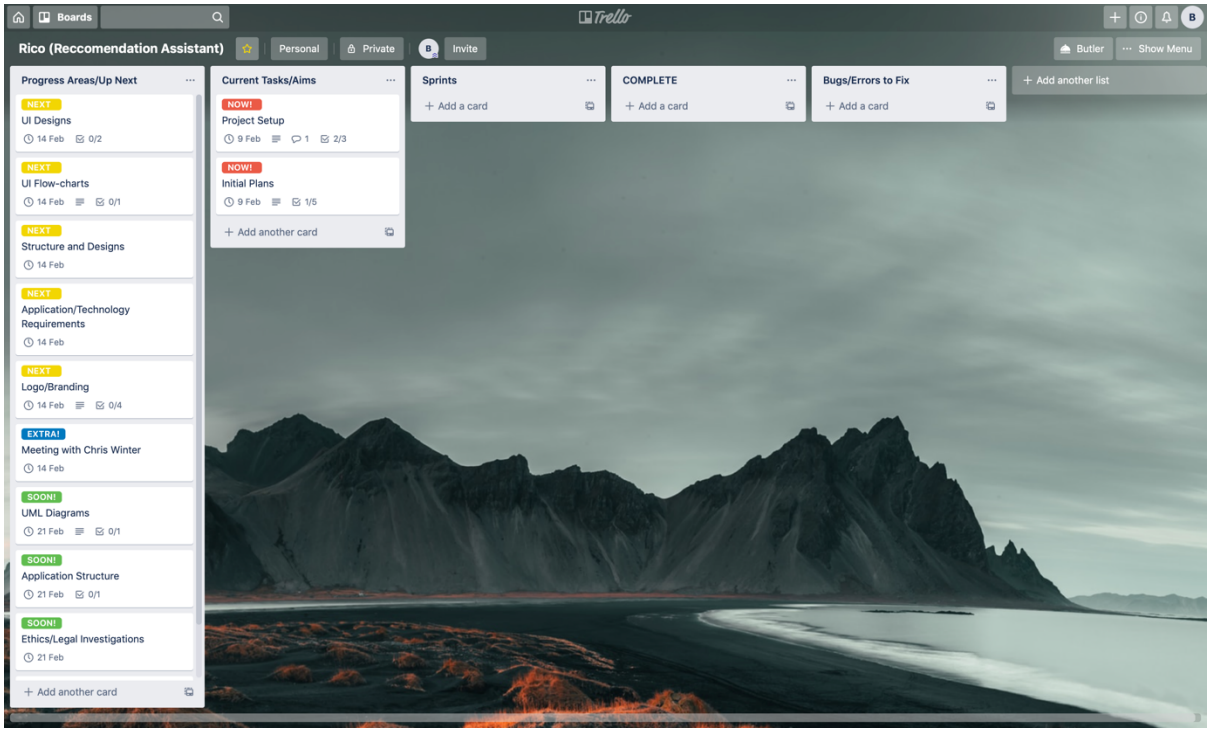
To add a movie recommendation to the database, navigate to add recommendations through Rico settings. Fill in all the fields and click on add movie.

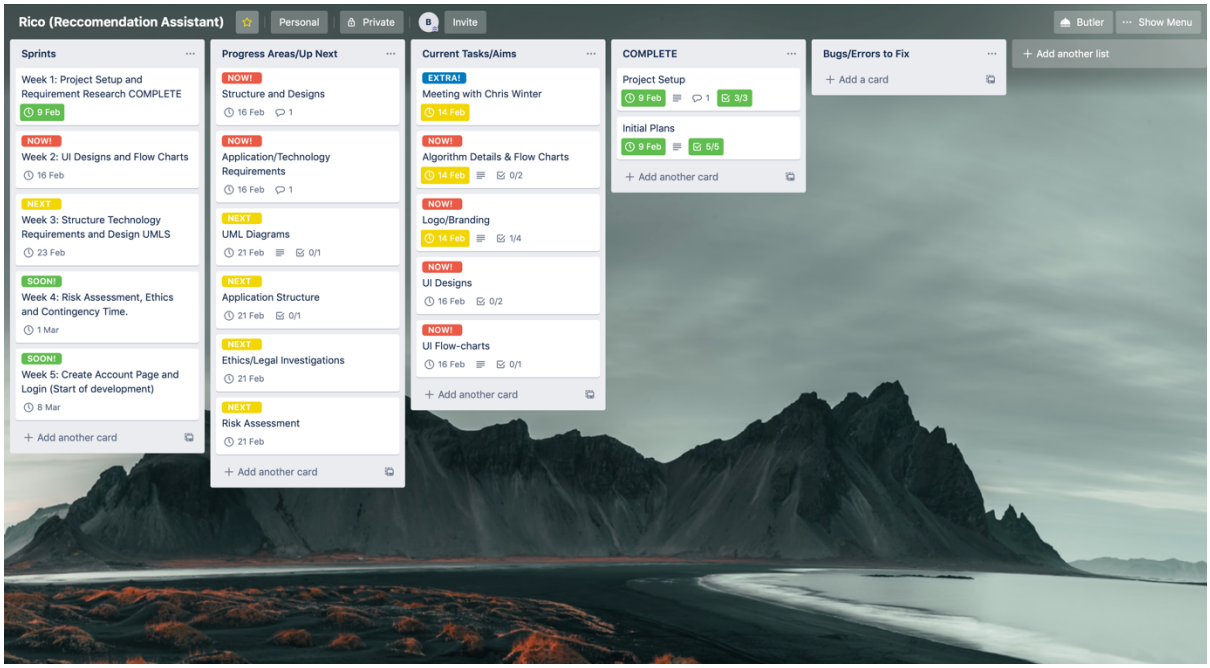


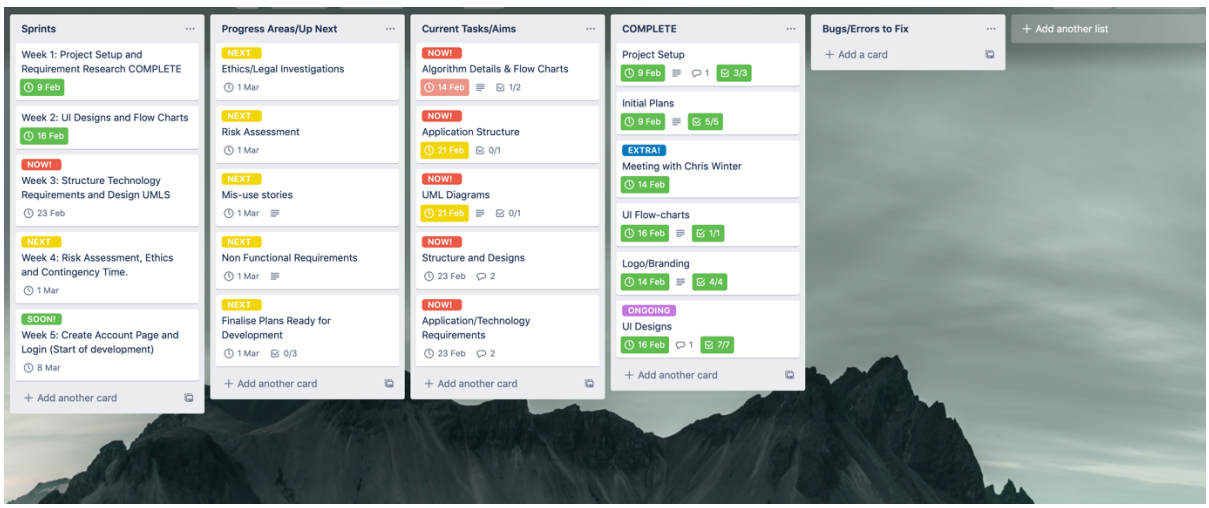
13.2 Trello Boards (Product Backlog)

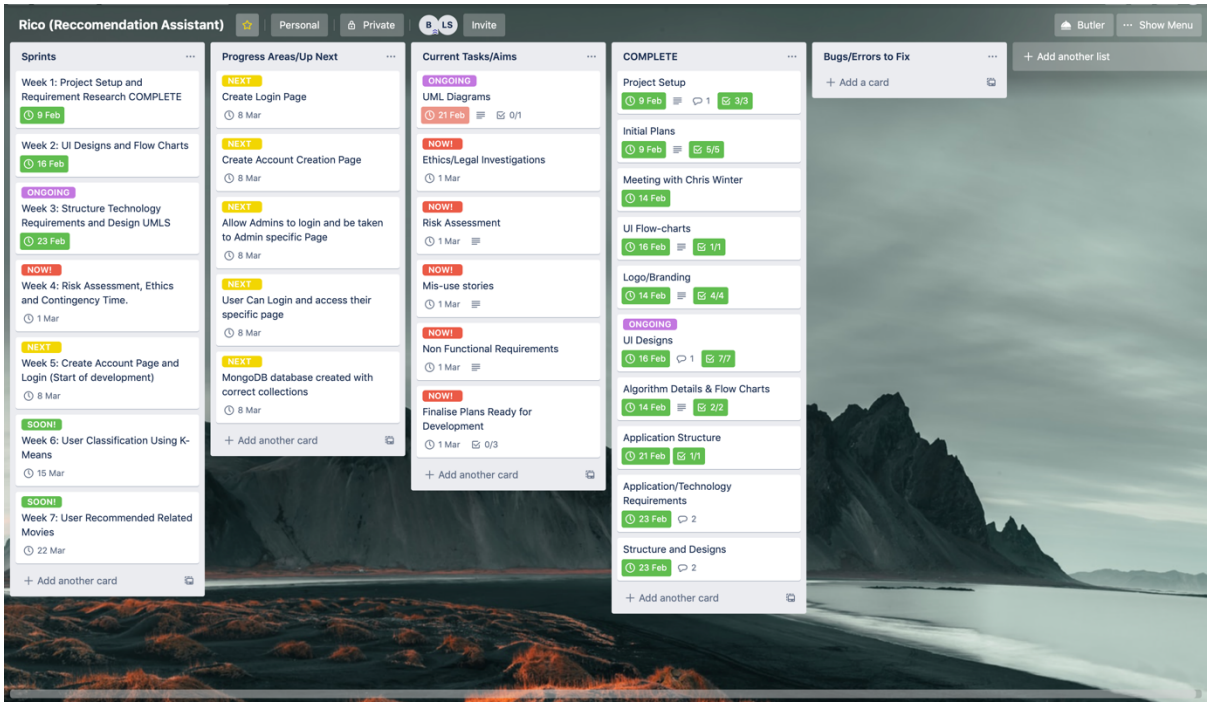
In chronological order.

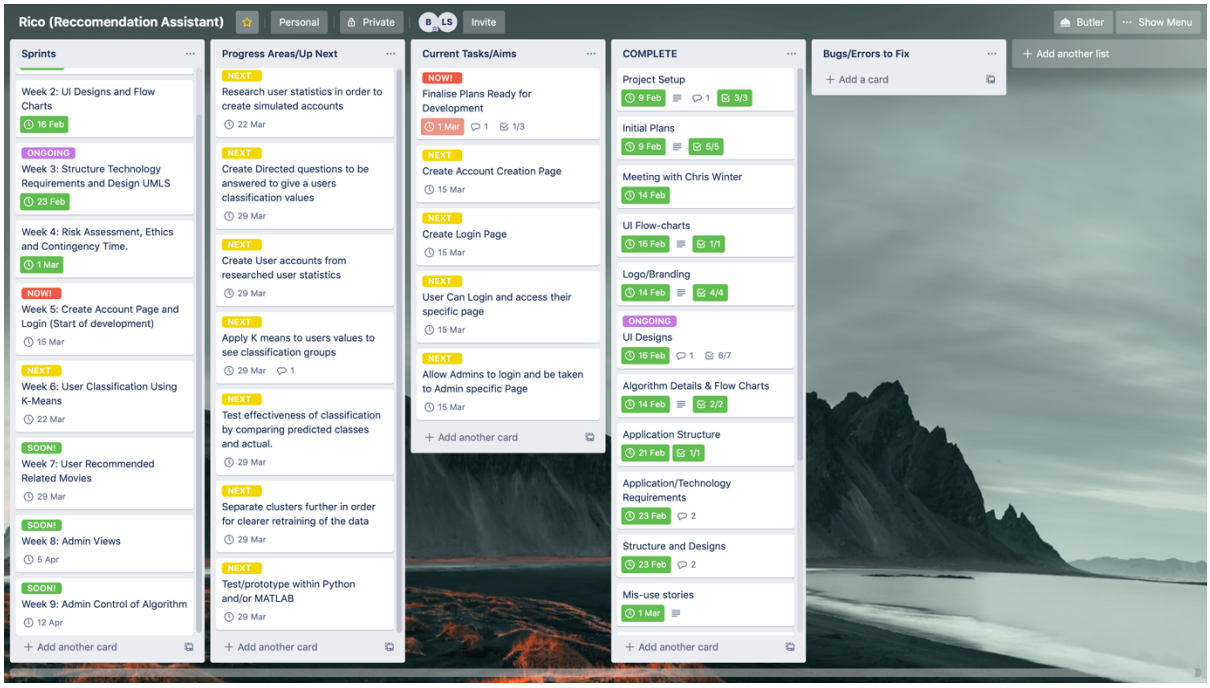


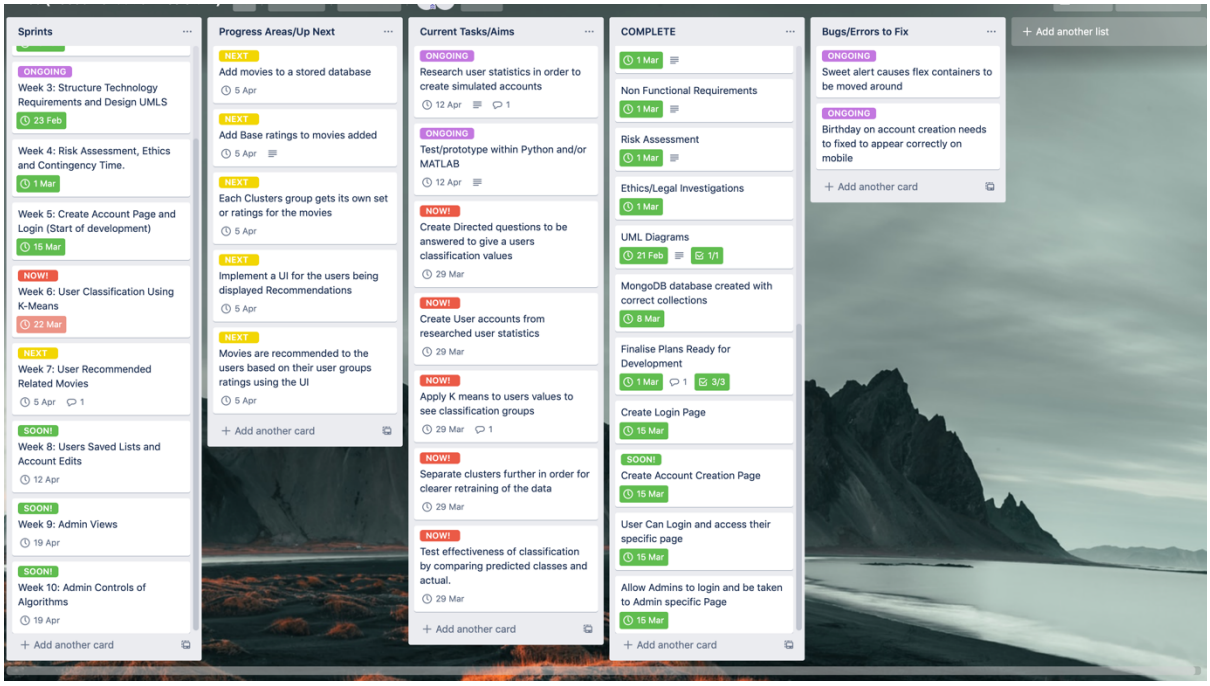


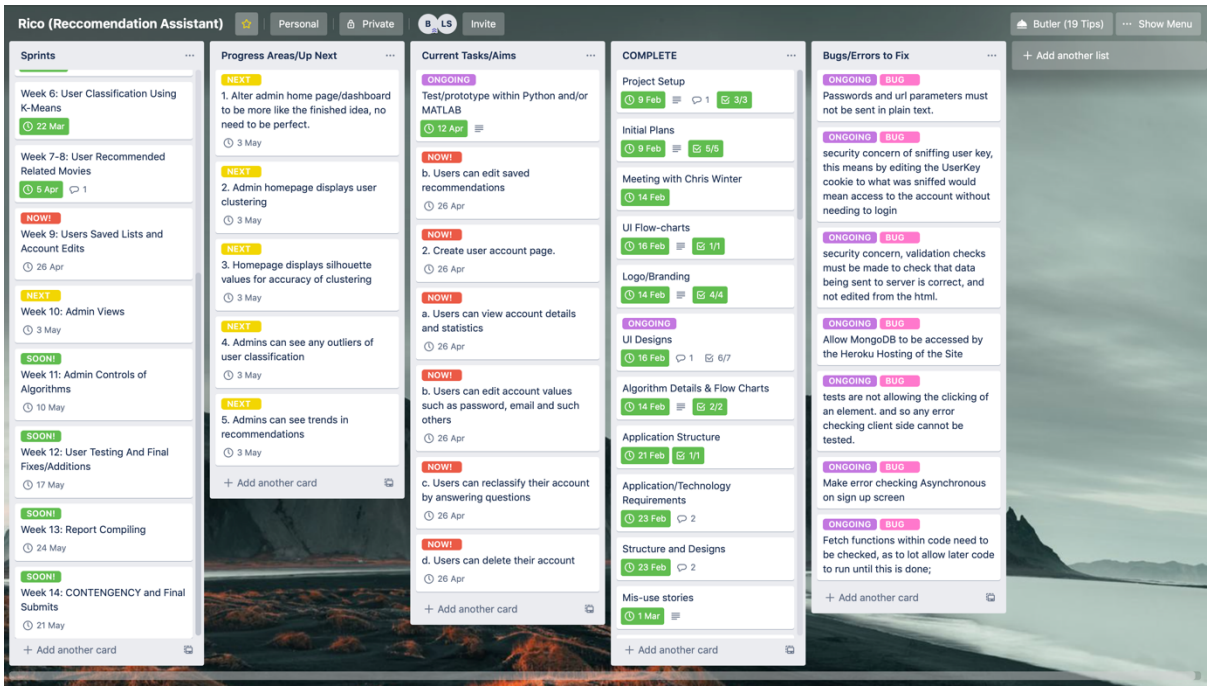

Rico (Reccomendation Assistant)
Personal
Private
Invite
Butler
Show Menu
Sprints
Week 1: Project Setup and Requirement Research COMPLETE
9 Feb
NOW!
Week 2: UI Designs and Flow Charts
16 Feb
NEXT
Week 3: Structure Technology Requirements and Design UMLS
23 Feb
SOON!
Week 4: Risk Assessment, Ethics and Contingency Time.
1 Mar
SOON!
Week 5: Create Account Page and Login (Start of development)
8 Mar
Add another card
Progress Areas/Up Next
NOW!
Structure and Designs
16 Feb
NOW!
Application/Technology Requirements
16 Feb
NEXT
UML Diagrams
21 Feb
0/1
NEXT
Application Structure
21 Feb
0/1
NEXT
Ethics/Legal Investigations
21 Feb
NEXT
Risk Assessment
21 Feb
Add another card
Current Tasks/Aims
EXTRA!
Meeting with Chris Winter
14 Feb
NOW!
Algorithm Details & Flow Charts
14 Feb
0/2
NOW!
Logo/Branding
14 Feb
1/4
NOW!
UI Designs
16 Feb
0/2
NOW!
UI Flow-charts
16 Feb
0/1
Add another card
COMPLETE
Project Setup
9 Feb
1
3/3
Initial Plans
9 Feb
5/5
Add another card
Bugs/Errors to Fix
Add a card
Add another list

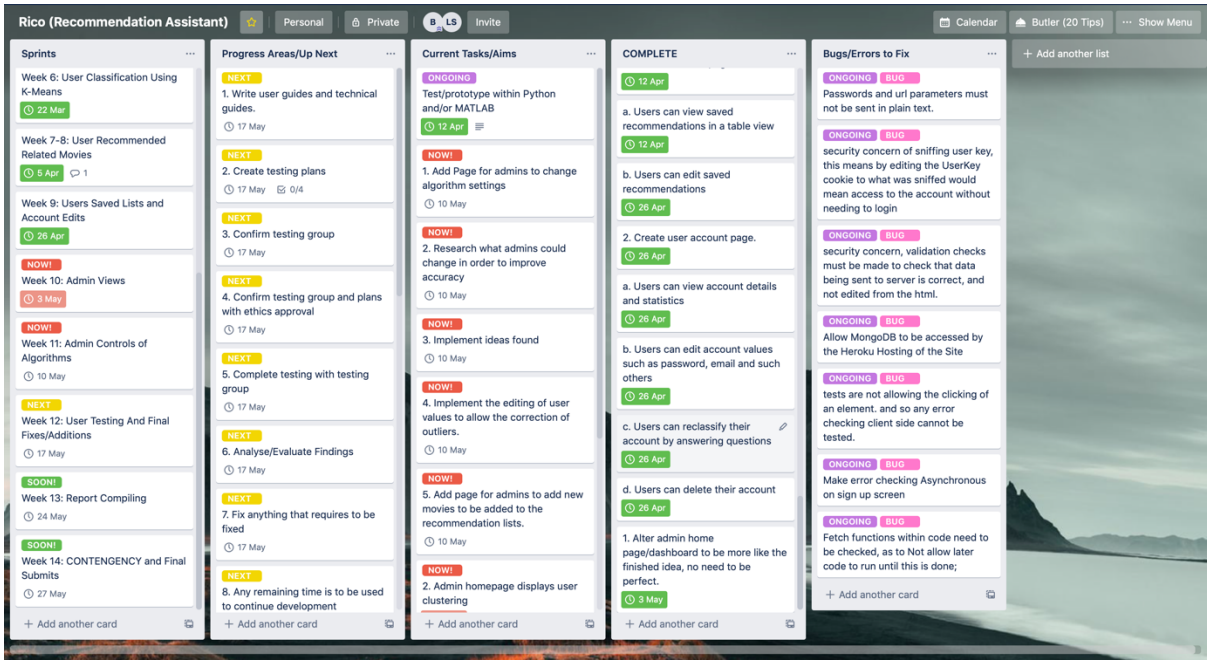

Sprints
Week 1: Project Setup and Requirement Research COMPLETE
9 Feb
Week 2: UI Designs and Flow Charts
16 Feb
NOW!
Week 3: Structure Technology Requirements and Design UMLS
23 Feb
NEXT
Week 4: Risk Assessment, Ethics and Contingency Time.
1 Mar
SOON!
Week 5: Create Account Page and Login (Start of development)
8 Mar
Add another card
Progress Areas/Up Next
NEXT
Ethics/Legal Investigations
1 Mar
NEXT
Risk Assessment
1 Mar
NEXT
Mis-use stories
1 Mar
NEXT
Non Functional Requirements
1 Mar
NEXT
Finalise Plans Ready for Development
1 Mar
0/3
Add another card
Current Tasks/Aims
NOW!
Algorithm Details & Flow Charts
14 Feb
1/2
NOW!
Application Structure
21 Feb
0/1
NOW!
UML Diagrams
21 Feb
0/1
NOW!
Structure and Designs
23 Feb
2
NOW!
Application/Technology Requirements
23 Feb
2
Add another card
COMPLETE
Project Setup
9 Feb
1
3/3
Initial Plans
9 Feb
5/5
EXTRA!
Meeting with Chris Winter
14 Feb
UI Flow-charts
16 Feb
1/1
Logo/Branding
14 Feb
4/4
ONGOING
UI Designs
16 Feb
1
7/7
Add another card
Bugs/Errors to Fix
Add a card
Add another list

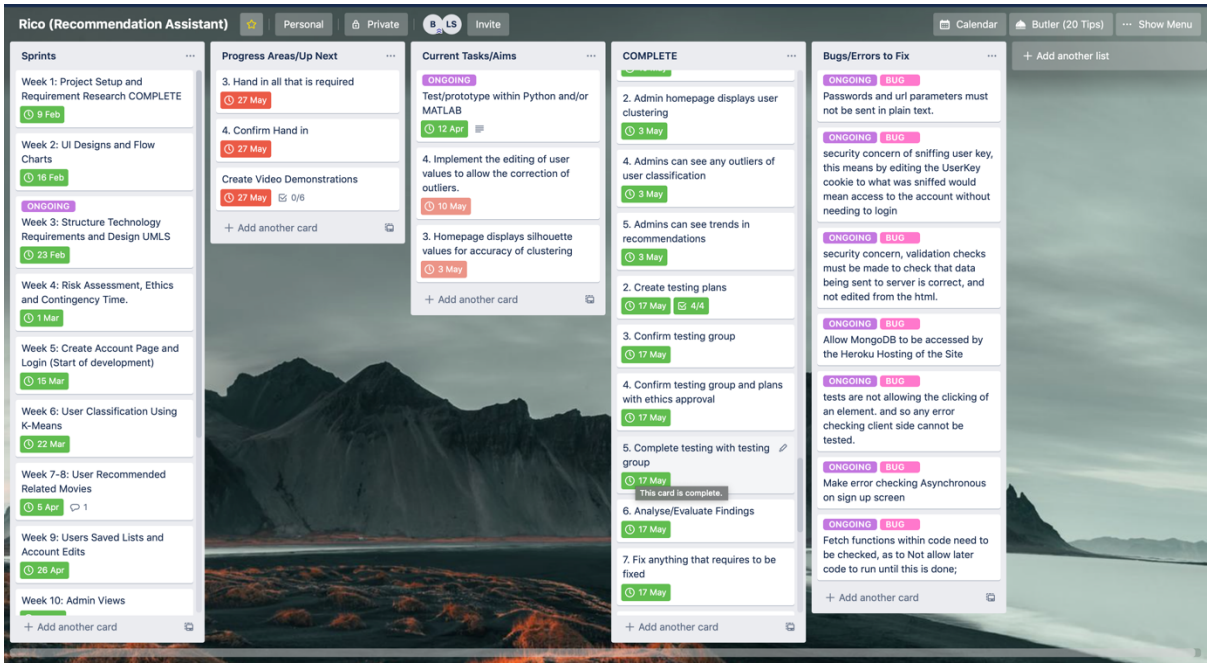

Rico (Reccomendation Assistant)
Personal
Private
Invite
Butler
Show Menu
Sprints
Week 1: Project Setup and Requirement Research COMPLETE
9 Feb
Week 2: UI Designs and Flow Charts
16 Feb
ONGOING
Week 3: Structure Technology Requirements and Design UMLS
23 Feb
NOW!
Week 4: Risk Assessment, Ethics and Contingency Time.
1 Mar
NEXT
Week 5: Create Account Page and Login (Start of development)
8 Mar
SOON!
Week 6: User Classification Using K-Means
15 Mar
SOON!
Week 7: User Recommended Related Movies
22 Mar
Add another card
Progress Areas/Up Next
NEXT
Create Login Page
8 Mar
NEXT
Create Account Creation Page
8 Mar
NEXT
Allow Admins to login and be taken to Admin specific Page
8 Mar
NEXT
User Can Login and access their specific page
8 Mar
NEXT
MongoDB database created with correct collections
8 Mar
Add another card
Current Tasks/Aims
ONGOING
UML Diagrams
21 Feb
0/1
NOW!
Ethics/Legal Investigations
1 Mar
NOW!
Risk Assessment
1 Mar
NOW!
Mis-use stories
1 Mar
NOW!
Non Functional Requirements
1 Mar
NOW!
Finalise Plans Ready for Development
1 Mar
0/3
Add another card
COMPLETE
Project Setup
9 Feb
3/3
Initial Plans
9 Feb
5/5
Meeting with Chris Winter
14 Feb
UI Flow-charts
16 Feb
1/1
Logo/Branding
14 Feb
4/4
ONGOING
UI Designs
16 Feb
7/7
Algorithm Details & Flow Charts
14 Feb
2/2
Application Structure
21 Feb
1/1
Application/Technology Requirements
23 Feb
Structure and Designs
23 Feb
Add another card
Bugs/Errors to Fix
Add a card
Add another list


Rico (Reccomendation Assistant)
Personal
Private
Invite
Butler
Show Menu
Sprints
Week 2: UI Designs and Flow Charts
16 Feb
ONGOING
Week 3: Structure Technology Requirements and Design UMLS
23 Feb
Week 4: Risk Assessment, Ethics and Contingency Time.
1 Mar
NOW!
Week 5: Create Account Page and Login (Start of development)
15 Mar
NEXT
Week 6: User Classification Using K-Means
22 Mar
SOON!
Week 7: User Recommended Related Movies
29 Mar
SOON!
Week 8: Admin Views
5 Apr
SOON!
Week 9: Admin Control of Algorithm
12 Apr
Add another card
Progress Areas/Up Next
NEXT
Research user statistics in order to create simulated accounts
22 Mar
NEXT
Create Directed questions to be answered to give a users classification values
29 Mar
NEXT
Create User accounts from researched user statistics
29 Mar
NEXT
Apply K means to users values to see classification groups
29 Mar
1
NEXT
Test effectiveness of classification by comparing predicted classes and actual.
29 Mar
NEXT
Separate clusters further in order for clearer retraining of the data
29 Mar
NEXT
Test/prototype within Python and/or MATLAB
29 Mar
Add another card
Current Tasks/Aims
NOW!
Finalise Plans Ready for Development
1 Mar
1
1/3
NEXT
Create Account Creation Page
15 Mar
NEXT
Create Login Page
15 Mar
NEXT
User Can Login and access their specific page
15 Mar
NEXT
Allow Admins to login and be taken to Admin specific Page
15 Mar
Add another card
COMPLETE
Project Setup
9 Feb
3/3
Initial Plans
9 Feb
5/5
Meeting with Chris Winter
14 Feb
UI Flow-charts
16 Feb
1/1
Logo/Branding
14 Feb
4/4
ONGOING
UI Designs
16 Feb
1
6/7
Algorithm Details & Flow Charts
14 Feb
2/2
Application Structure
21 Feb
1/1
Application/Technology Requirements
23 Feb
2
Structure and Designs
23 Feb
2
Mis-use stories
1 Mar
Add another card
Bugs/Errors to Fix
Add a card
Add another list


Sprints
ONGOING
Week 3: Structure Technology Requirements and Design UMLS
23 Feb
Week 4: Risk Assessment, Ethics and Contingency Time.
1 Mar
Week 5: Create Account Page and Login (Start of development)
15 Mar
NOW!
Week 6: User Classification Using K-Means
22 Mar
NEXT
Week 7: User Recommended Related Movies
5 Apr
SOON!
Week 8: Users Saved Lists and Account Edits
12 Apr
SOON!
Week 9: Admin Views
19 Apr
SOON!
Week 10: Admin Controls of Algorithms
19 Apr
Add another card
Progress Areas/Up Next
NEXT
Add movies to a stored database
5 Apr
NEXT
Add Base ratings to movies added
5 Apr
NEXT
Each Clusters group gets its own set or ratings for the movies
5 Apr
NEXT
Implement a UI for the users being displayed Recommendations
5 Apr
NEXT
Movies are recommended to the users based on their user groups ratings using the UI
5 Apr
Add another card
Current Tasks/Aims
ONGOING
Research user statistics in order to create simulated accounts
12 Apr
ONGOING
Test/prototype within Python and/or MATLAB
12 Apr
NOW!
Create Directed questions to be answered to give a users classification values
29 Mar
NOW!
Create User accounts from researched user statistics
29 Mar
NOW!
Apply K means to users values to see classification groups
29 Mar
NOW!
Separate clusters further in order for clearer retraining of the data
29 Mar
NOW!
Test effectiveness of classification by comparing predicted classes and actual.
29 Mar
Add another card
COMPLETE
1 Mar
Non Functional Requirements
1 Mar
Risk Assessment
1 Mar
Ethics/Legal Investigations
1 Mar
UML Diagrams
21 Feb
1/1
MongoDB database created with correct collections
8 Mar
Finalise Plans Ready for Development
1 Mar
3/3
Create Login Page
15 Mar
SOON!
Create Account Creation Page
15 Mar
User Can Login and access their specific page
15 Mar
Allow Admins to login and be taken to Admin specific Page
15 Mar
Add another card
Bugs/Errors to Fix
ONGOING
Sweet alert causes flex containers to be moved around
ONGOING
Birthday on account creation needs to fixed to appear correctly on mobile
Add another card
Add another list


Rico (Reccomendation Assistant)
Personal
Private
Invite
Butler (19 Tips)
Show Menu
Sprints
Week 6: User Classification Using K-Means
22 Mar
Week 7-8: User Recommended Related Movies
5 Apr
NOW!
Week 9: Users Saved Lists and Account Edits
26 Apr
NEXT
Week 10: Admin Views
3 May
SOON!
Week 11: Admin Controls of Algorithms
10 May
SOON!
Week 12: User Testing And Final Fixes/Additions
17 May
SOON!
Week 13: Report Compiling
24 May
SOON!
Week 14: CONTENGENCY and Final Submits
21 May
Add another card
Progress Areas/Up Next
NEXT
1. Alter admin home page/dashboard to be more like the finished idea, no need to be perfect.
3 May
NEXT
2. Admin homepage displays user clustering
3 May
NEXT
3. Homepage displays silhouette values for accuracy of clustering
3 May
NEXT
4. Admins can see any outliers of user classification
3 May
NEXT
5. Admins can see trends in recommendations
3 May
Add another card
Current Tasks/Aims
ONGOING
Test/prototype within Python and/or MATLAB
12 Apr
NOW!
b. Users can edit saved recommendations
26 Apr
NOW!
2. Create user account page.
26 Apr
NOW!
a. Users can view account details and statistics
26 Apr
NOW!
b. Users can edit account values such as password, email and such others
26 Apr
NOW!
c. Users can reclassify their account by answering questions
26 Apr
NOW!
d. Users can delete their account
26 Apr
Add another card
COMPLETE
Project Setup
9 Feb
3/3
Initial Plans
9 Feb
5/5
Meeting with Chris Winter
14 Feb
UI Flow-charts
16 Feb
1/1
Logo/Branding
14 Feb
4/4
ONGOING
UI Designs
16 Feb
6/7
Algorithm Details & Flow Charts
14 Feb
2/2
Application Structure
21 Feb
1/1
Application/Technology Requirements
23 Feb
2
Structure and Designs
23 Feb
2
Mis-use stories
1 Mar
Add another card
Bugs/Errors to Fix
ONGOING BUG
Passwords and url parameters must not be sent in plain text.
ONGOING BUG
security concern of sniffing user key, this means by editing the UserKey cookie to what was sniffed would mean access to the account without needing to login
ONGOING BUG
security concern, validation checks must be made to check that data being sent to server is correct, and not edited from the html.
ONGOING BUG
Allow MongoDB to be accessed by the Heroku Hosting of the Site
ONGOING BUG
tests are not allowing the clicking of an element. and so any error checking client side cannot be tested.
ONGOING BUG
Make error checking Asynchronous on sign up screen
ONGOING BUG
Fetch functions within code need to be checked, as to lot allow later code to run until this is done;
Add another card
Add another list


Rico (Recommendation Assistant)
Personal
Private
Invite
Calendar
Butler (20 Tips)
Show Menu
Sprints
Week 6: User Classification Using K-Means
22 Mar
Week 7-8: User Recommended Related Movies
5 Apr
Week 9: Users Saved Lists and Account Edits
26 Apr
NOW!
Week 10: Admin Views
3 May
NOW!
Week 11: Admin Controls of Algorithms
10 May
NEXT
Week 12: User Testing And Final Fixes/Additions
17 May
SOON!
Week 13: Report Compiling
24 May
SOON!
Week 14: CONTENGENCY and Final Submits
27 May
Add another card
Progress Areas/Up Next
NEXT
1. Write user guides and technical guides.
17 May
NEXT
2. Create testing plans
17 May
0/4
NEXT
3. Confirm testing group
17 May
NEXT
4. Confirm testing group and plans with ethics approval
17 May
NEXT
5. Complete testing with testing group
17 May
NEXT
6. Analyse/Evaluate Findings
17 May
NEXT
7. Fix anything that requires to be fixed
17 May
NEXT
8. Any remaining time is to be used to continue development
Add another card
Current Tasks/Aims
ONGOING
Test/prototype within Python and/or MATLAB
12 Apr
NOW!
1. Add Page for admins to change algorithm settings
10 May
NOW!
2. Research what admins could change in order to improve accuracy
10 May
NOW!
3. Implement ideas found
10 May
NOW!
4. Implement the editing of user values to allow the correction of outliers.
10 May
NOW!
5. Add page for admins to add new movies to be added to the recommendation lists.
10 May
NOW!
2. Admin homepage displays user clustering
Add another card
COMPLETE
12 Apr
a. Users can view saved recommendations in a table view
12 Apr
b. Users can edit saved recommendations
26 Apr
2. Create user account page.
26 Apr
a. Users can view account details and statistics
26 Apr
b. Users can edit account values such as password, email and such others
26 Apr
c. Users can reclassify their account by answering questions
26 Apr
d. Users can delete their account
26 Apr
1. Alter admin home page/dashboard to be more like the finished idea, no need to be perfect.
3 May
Add another card
Bugs/Errors to Fix
ONGOING
BUG
Passwords and url parameters must not be sent in plain text.
ONGOING
BUG
security concern of sniffing user key, this means by editing the UserKey cookie to what was sniffed would mean access to the account without needing to login
ONGOING
BUG
security concern, validation checks must be made to check that data being sent to server is correct, and not edited from the html.
ONGOING
BUG
Allow MongoDB to be accessed by the Heroku Hosting of the Site
ONGOING
BUG
tests are not allowing the clicking of an element. and so any error checking client side cannot be tested.
ONGOING
BUG
Make error checking Asynchronous on sign up screen
ONGOING
BUG
Fetch functions within code need to be checked, as to Not allow later code to run until this is done;
Add another card
Add another list


Rico (Recommendation Assistant)
Personal
Private
Invite
Calendar
Butler (20 Tips)
Show Menu
Sprints
Week 1: Project Setup and Requirement Research COMPLETE
9 Feb
Week 2: UI Designs and Flow Charts
16 Feb
ONGOING
Week 3: Structure Technology Requirements and Design UMLS
23 Feb
Week 4: Risk Assessment, Ethics and Contingency Time.
1 Mar
Week 5: Create Account Page and Login (Start of development)
15 Mar
Week 6: User Classification Using K-Means
22 Mar
Week 7-8: User Recommended Related Movies
5 Apr
Week 9: Users Saved Lists and Account Edits
26 Apr
Week 10: Admin Views
Add another card
Progress Areas/Up Next
3. Hand in all that is required
27 May
4. Confirm Hand in
27 May
Create Video Demonstrations
27 May
0/6
Add another card
Current Tasks/Aims
ONGOING
Test/prototype within Python and/or MATLAB
12 Apr
4. Implement the editing of user values to allow the correction of outliers.
10 May
3. Homepage displays silhouette values for accuracy of clustering
3 May
Add another card
COMPLETE
2. Admin homepage displays user clustering
3 May
4. Admins can see any outliers of user classification
3 May
5. Admins can see trends in recommendations
3 May
2. Create testing plans
17 May
4/4
3. Confirm testing group
17 May
4. Confirm testing group and plans with ethics approval
17 May
5. Complete testing with testing group
17 May
This card is complete.
6. Analyse/Evaluate Findings
17 May
7. Fix anything that requires to be fixed
17 May
Add another card
Bugs/Errors to Fix
ONGOING
BUG
Passwords and url parameters must not be sent in plain text.
ONGOING
BUG
security concern of sniffing user key, this means by editing the UserKey cookie to what was sniffed would mean access to the account without needing to login
ONGOING
BUG
security concern, validation checks must be made to check that data being sent to server is correct, and not edited from the html.
ONGOING
BUG
Allow MongoDB to be accessed by the Heroku Hosting of the Site
ONGOING
BUG
tests are not allowing the clicking of an element. and so any error checking client side cannot be tested.
ONGOING
BUG
Make error checking Asynchronous on sign up screen
ONGOING
BUG
Fetch functions within code need to be checked, as to Not allow later code to run until this is done;
Add another card
Add another list

13.3 Sprint Planning Document

Week 1: Project Start-up and Requirements Analysis

Start Date: 3rd February - End Date: 9th February

Goal

All things that are needed for this project are set up by the end of this week, this includes GitHub, Trello and the vital requirement documents (User Stories, Functional Requirements, Project outline, Basic Gannt Chart).

User Stories to Base Development From.

- N/A

Related Functional Requirements

- N/A

Steps To Achieve Goal:

1. Create project outline
2. Create Functional Requirements
3. Create User Stories
4. Order requirements into core, desirable and additional.
5. Create a basic Gannt chart.
6. Look into market research, similar products.
7. Review.

Sprint Review

Everything this week has been completed on time. The functional requirements are complete and cover everything that can be foreseen at this stage of planning, however, this may change throughout development. With these plans user stories were generated as to bring the functional requirements into the users grasp as to ensure that every stage of development is taking the user into mind with what they will actually need the software for. The project outline was fill by using the initial project vision document as this proposal covers everything this software aims to solve. The Gannt chart has been left to complete until more sprint plans are generated as each week of planning can be given user stories to fill with points to allocate the time needed each week on each task. Overall a solid start, although, it must be noted that these initial plans are not the set-in stone requirements of the project, as with any project better ideas or pathways for development may emerge and may be taken.

Week 2: UI Designs and UI Flow-Charts

Start Date: 10th February - End Date: 16th February

Goal

This week UI designs and UI flow charts are to be created. From lasts weeks planning we should be able to see what the UI needs to allow the user to be able to perform. Also, along with the UI designing the logo and branding of the project can be outlined.

User Stories to Base Development From.

- ALL

Related Functional Requirements

- ALL

Steps To Achieve Goal:

1. Generate UI designs for all pages required (Mobile and Desktop) following HCI principles. Refer to user stories and functional requirements for what each page is required to allow access to.
    a. Login screen.
    b. Account Creation.
    c. Home page (Area of recommendation selections).
    d. Recommendation page for specific area.
    e. ‘My Account’ page
    f. Admin specific page (Data and analytics)
    g. Admin Controls page.
2. Generate UI flow charts to outline how all the above pages allow access to each other.
3. Create potential logos and other branding of the project.
    a. Logo
    b. Name
    c. Slogan
    d. Colour Scheme (Should be clear from UI Designs).
4. Create Algorithm plans ready for next week.
    a. Algorithm outline and basic flowchart for algorithm flow

Sprint Review

Overall this week went okay, as for plans being complete everything is great, however items such UI designs could do with a cognitive walkthrough as to test its HCI effectiveness. All flowcharts are fantastic for development as they outline the intended flow of each aspect of the software, these will be used extensively in order to aid development and at the end of development to be certain that the planned project goals have been achieved. UI designs have been marked as ongoing as they may change as development occurs, also, this way when planning to implement each

item for each weekly sprint they can be re-drawn if needed to be more targeted to the weekly sprint goals. This week's goal has still been achieved overall.

Week 3: Structure, Technology Requirements and Design UMLS

Start Date: 17th February - End Date: 23rd February

Goal

The goal for this week is to outline the backend of the software and all its technology requirements. Diagrams should be produced in order to better describe and show intentions of technology plans. If in the case of multiple approaches being discovered, each approach should have its pros and cons outlined to aid the final decision on approach.

User Stories To Base Development From.

- ALL

Related Functional Requirements

- ALL

Steps To Achieve Goal:

1. Create technology stack diagrams
2. Create a List of Technology Requirements
3. Outline what is data is going to be stored and the flow (Data Review)
4. Initial UML plans
5. Speak to technical advisors for advice and approval of technology stack design.

Sprint Review

This week proved difficult due to illness and problems with housing and maintenance. Some tasks have resulted being moved into contingency time such as getting approval and advise from technical advisors and the project supervisor. However, as for producing the diagrams things are on track and complete. The things to note are that these plans produced from this week are open to change throughout development, but still, they outline what is expected from the software. as long as the goals set out from the plans are still achieved there isn't a cause for concern. The goal from this sprint has been achieved however the checks and approval is yet to occur, development can still start but an awareness that certain aspects may change if the approach outlined by previous plans is altered

A Meeting for advice was conducted with the IT professional Chris Winter, a previous IBM employee with several years of experience within the industry. In this meeting the idea of non-functional requirements and misuse cases was brought to light. Including these items with the plans already made we can focus attention to the user even further in order to ensure user satisfaction is achieved. The time that will be

used for this will be next week's contingency time. This meeting also addresses the idea of including medical recommendations in the software, and that it would be wise to focus on getting the software working with Movies first, then activities and leave the software open to including medical advice. This is due to the difficulty in getting accurate medical data that isn't linked to patients as well as the ethical problems of giving medical advice that is not backed by a medical professional.

Week 4: Risk Assessment, Ethics and Contingency time.

Start Date: 24th February - End Date: 1st March

Goal

This week, as all previous plans should have been completed by this point in time, an extensive risk assessment plan can be produced. All risks that could potentially occur should be outlined as well as what precautions should be made in order to minimise their risk, in addition to this a risk action plan should be put into place just in case the risk occurs as to be prepared for the worst. Ethical investigations need to be performed as to ensure that the software complies to legal and ethical regulations set out by law and the university. Finally, the Gannt chart can be updated in order to accommodate each user story, each user story should be awarded points in order to allocate time available each week to each user story/task.

User Stories To Base Development From.

- N/A

Related Functional Requirements

- N/A

Steps To Achieve Goal:

1. Ethical and legal investigations and research.
2. Risk assessment.
3. Update Gannt chart to use user points.
4. Contingency time for unfinished and additional tasks.
    a. Mis-use user stories (Chris Winter meeting).
    b. Non-functional Requirements (Chris Winter meeting).
    c. Finish any unfinished tasks from last week.
5. Signoff plans
    a. Meeting with technical advisors and project supervisor to confirm the start of development.

Sprint Review

This week finalised plans, although not all plans initially thought of were completed, the ones started that required contingency were now signed off and complete. However, I am satisfied that the plans completed are more than sufficient in the outlining on what Is expected from the software. Having the meeting with

David Walker (Technical Advisor) proved that understanding of the whole project and how everything is linked is possible when looking at the plans produced. It must be made certain that throughout development, we constantly refer back to plans at several stages, especially at every sprint review in order to ensure nothing core and vital are missed out in the project submitted.

Week Away – Due to Other Work

Start Date: 2nd March - End Date: 8th March

Goal

No development is to take place this week, due to arrangements with personal life it will prove difficult to produce any targeted goals due to the uncertainty of available time. 2 weeks were planned as time off over Easter, however this has been split into 2 separate weeks as to allow this time here to be free.

User Stories To Base Development From.

- N/A

Related Functional Requirements

- N/A

Steps To Achieve Goal:

- N/A

Sprint Review

Nothing to review as minimal work was completed this week, however, considering there was plan for no work to take place this week, any is a bonus.

Week 5: Create Account Page and Login (Start of development).

Start Date: 9th March - End Date: 15th March

Goal

User is greeted with a login screen on the loading of the application, a signup page will available in order to create a new account. HCI principles and basic security such as encryption and checking user privileges should be implemented. There should only be one centralised login for users and admins.

User Stories To Base Development From.

- I as an admin need to be able to login, so I can fulfil my administrator duties so that I can help the users of the software.

- I as the user of the software, would like to log into my account and be able to access my recommendations, saved or new.

- - I as a user of the software, would like navigation of the application to be easy and clean, automatic redirects to where I need to be would assist in this. This is so I am always at the right place at the right time.

Related Functional Requirements

- *User Account Creation Page* **(CORE)**
  - *User account details are entered (Username, Name, Password, Email).*
  - *User is guided through set questions (About Me), in order to find their classification.*
  - *Details are run through classification algorithm and the user's classification ID is stored with their account.*
  - There is no link between their account and data, once data is condensed into 3 dimensions, their input data is deleted. (see user classification).
  - *User account is created and stored, user is redirected to login (to validate details) or automatically logged in?*

- *When accessing the app, if not logged in this is the page that will greet the user.* **(CORE)**
  - *If a user tries to manually access page, they do not have access to they will be redirected to the last page they were on/login if they were not logged in.*

- *User can use their login details and be redirected to the home page of the software where they can select the area that they want recommendations.* **(CORE)**
  - *Login page – this will be the home of the application if not logged in.*
  - *One user login for all users, whether admin or not (simplified process).*
  - *Any access to a page that requires you to be logged in will redirect here.*

- *User can login using their details and be navigated into the application.*
- *User can be navigated to the create account page if they do not have an account.*
- *Admin accounts cannot be created, they must be created by another admin/developer.*

ASPECT: Home Page

- *User can navigate to available pages from here (Account, logout, help, recommendation pages, etc.).* **(CORE)**
  - *Navigation bar at the top to navigate to different pages*
  - *Navigation bar at the left in order to navigate to different recommendation areas*

- Clean and simple input of user data **(CORE)**
  - Users are never asked too complex of questions to answer.
  - Users data is cleansed upon input.

Steps To Achieve Goal:

1. Setup MEAN stack ready to base development from.
2. Set up continuous integration pipeline
   a. Implement templates for testing and server side npm tests
   b. Link GitHub to Travis CI
   c. Link Heroku (Hosting) to Github
   d. Test pipeline
3. Create signup page
   a. Draw from UI designs and create a responsive/reactive login page
   b. Ensure HCI design
   c. Allow Users to create account
4. Create login page
   a. Draw from UI designs and create a responsive/reactive login page
   b. Ensure HCI design
   c. Allow user to login with created details
   d. Depending on users rank (admin or user), they are taken to their relating homepage.

Sprint Review

Development started off slow and this sprint run slightly longer than scheduled into next weeks allotted time. Development was slow as is was not assumed how much work and time goes into setting up the development pipeline and necessary environments in order for this web application to operate. Everything that needed to be implemented this week was completed with relative ease, however, the problems that arose during development was not taken into account. This put this week behind in addition with implementing non-core features, too much focus was put into

security and HCI principles that caused too many additional items to be developed this week ultimately making this sprint last longer than it should have.

In the end, this sprint was achieved, and development has gone far further than planned and this is great, however, it must be remembered that sticking to plans must come first, as to achieve a minimum viable product core functionality is priority. If during a sprint extra time is found, then extra development can occur. Until we are back on track, core functionality must be implemented first and moving onto sprints on time is the focus from here onwards. Once we are back on track or ahead development like this week can be continued, as it has made the overall look and feel of the application professional; too high of a standard has been set thus far.

Week 6: User Classification using K means

Start Date: 16th March - End Date: 22nd March

Goal

When a user creates an account/on first login/ no classification details are on record, a user is asked questions that will classify them into a cluster/classification group that will allow them to share moving ratings for a group of similar users.

User Stories To Base Development From.

- I as a user of the software want to be able to be classified into a group of similar users so that I can be recommended items that are popular and well suited to those like me.

Related Functional Requirements

- *User Classification (K-means prototype (Neural Network Later))* **(CORE)**
    - *User inputs data based on questions asked in order to locate and classify themselves to similar users.*
    - From these questions, 3 values are calculated as to condense the input data into workable data. Input data is deleted/not stored only their calculated values. (See Security).
    - *These 3 calculated values are plotted using K-means clustering in order to find a user's similar user group/cluster.*
    - *User is given their cluster ID and sorted with their account details.*
    - *User can run a reclassification at any time in case of a change of interests / recommendations are not 100%.*

Steps To Achieve Goal:

1. Create research into what can classify users relating to movie preferences
    a. Research movie genres and their audience
    b. Brainstorm possible relations and possible values to be calculated through questions

    c. Create questions that will allow users to give accurate answers that lead to values to be used for classification
2. Prototype findings using MATLAB/Python in order plot current K-Means cluster classification.
    a. Plot user values and analyse clusters to test effectiveness or questions. (create simulated user values from previous statistical reports.)
    b. Adjust questions/values in order to get accurate clustering of users
3. Apply a further function to the data in order to further distance the clusters
4. Translate K-Means and questions into application code.
    a. Create a form to enter questions
    b. Create users that match the simulated user accounts from prototyping.
    c. Allow newly created users to be classified and their classifications to be stored.

Sprint Review

User Classification took longer than expected. Problems occurred throughout development such as planned ideas for the implementation of algorithms not working as first thought. The problems were items such as theorising an accurate method of classifying a user by two values. On top of this how to classify movies based on two values x and y, several prototypes were drawn up and the working version edited to be accurate. In order to place a user into a cluster a custom algorithm was also created, and this took far more time than first planned.

Overall everything ended up complete in a comfortable state, furthermore by focusing on the minimum viable product, progress and advancements was kept close to the project's needs. The test users in the isolated test environment seem so far to be working perfectly, this will be tested further when user testing occurs.

NOTE: instead of using the planned tensorflow.js, the node package 'skmeans' allowed for a minimal version of K-means to be used rather than the overkill that would have been tensorflow.js. In order to test that this node package was working as required, data produced by 'skmeans' was compared with the prototype created in MATLAB, all from the same test data.

Week 7-8: User Recommended Related Movies

Start Date: 23rd March - End Date: 5th April

Goal

Users, when logged in, can go through a visual stack of recommended movies that users of similar statistics (matching cluster group) have rated highly. Best matches/highest rated in that cluster group to be displayed at the top of the stack. User can rate yes or no for the recommendation or say to save to favourites.

User Stories To Base Development From.

CORE

- I as the user would like to be able to navigate to all available pages with ease, this is so I can access all items that I want from the software, such as my account details and my saved recommendations.

- I as a user, would like to view my saved recommendations that I have selected to save, this is to save me time from getting my favourite/most suitable items re-recommended through the recommendation algorithm.

- I as the user would like to remove recommendations from my saved stack, I can select whether they are put back ready to be re-recommended or to not be presented with them again. This is so I can manage my list of saved recommendations.

- I as a user, would like to be suggested recommendations for my chosen area that are most suitable for myself and users similar to me, this is because this is the main idea of using this software.

- I as a user would like to be able to reject recommendations to not be shown again as they are not suitable to me, this should also rate it to be bad for similar users to me, this is so I am not shown results like this again as it is not for me.

- I as a user, would like to save good recommendations, this will mean more recommendations in relation to the one saved will appear, this is so I can be recommended more accurate items in the future.

DESIRABLE

- I as a user would like to say that a recommendation is good, but not for me right now, this will increase the chance of this item popping up again, but it will not be saved.

Related Functional Requirements

- *Recommendations are suggested to the user based on similar users interests to them.* **(CORE)**
    - *All recommendations are based on how people in the same classification of the user has responded to each recommendation.*

    - *Most suitable recommendations are shown first, then as the user goes through the stack of recommendations, they will decrease in match percentage*

- *User, through the use of the software, will give feedback on if the recommendation was good for them or not.* **(CORE)**
    - *A user saying yes to the recommendation will increase the suitability rating average for that classification/cluster group. Saying no will decrease the rating average.*
    - *How the rating is given a value needs to be calculated based upon the algorithms used. By the majority of users in a cluster preferring that recommendation a rating closer to 100 should be used; closer to 0 for when the users of that cluster have a dislike/ rejection to the recommendation.*
- User can save recommendations to a list that they can view. **(CORE)**
    - *When saying yes to the recommendation the recommendation is saved to the list of saved recommendations.*

Steps To Achieve Goal:

1. Add movies and base ratings for all cluster groups to database.
2. Alter user home page to be more like the finished idea, no need to be perfect.
3. Create angular app to display recommendations in a stack of results
4. Allow users to rate recommendations
    a. User can reject recommendation (do not show again)
    b. User can like recommendation (give a match rating and this recommendation will display again)
    c. User can save recommendation (will be saved to their stack of matches for them, and return a 100% match)
    d. All ratings affect the overall average of match percentage for that group.

Sprint Review

From This Point on Sprint planning is a little different due to COVID-19, This Disrupted work in such a way that deadlines became late. In order to stay somewhat on track and to know the original deadlines this sprint document along with the Gannt chart stayed with original timings. This is so referral to where things should be could be made as to keep progress as close to the original deadlines as possible. Once one sprint was complete, progress onto the next immediately started.

This Week went extremely well, so much so that this sprint completed within 3 days. The use of the flow chart plans helped to visualise what exactly should be focused on and what each button needed to perform in order to fulfil this week's sprint goal. The angular application was skipped however, this could later be converted into an app very easily as this approach was taken baring this in mind; so, all functions and such were implemented ready to be copied and converted. For the

remainder of this sprints time period next week's sprint will be started in order to catch up on lost time due to COVID-19.

Week 9: Users Saved lists and Account Settings

Start Date: 6th April - End Date: 12th April

Goal

Users can view their account details and stats and change aspects such as password, email, and even delete their account. They can ask to re-classify their group and also if wanted, delete their account. Users also have the ability to view their saved list of recommendations and edit this list, such as reject/delete or return it to circulation of their recommendations.

User Stories To Base Development From.

CORE

- I as a user, would like to view my saved recommendations that I have selected to save, this is to save me time from getting my favourite/most suitable items re-recommended through the recommendation algorithm.

- I as the user would like to remove recommendations from my saved stack, I can select whether they are put back ready to be re-recommended or to not be presented with them again. This is so I can manage my list of saved recommendations.

- I as a user, would like to be suggested recommendations for my chosen area that are most suitable for myself and users similar to me, this is because this is the main idea of using this software.

- I as a user would like to be able to reject recommendations to not be shown again as they are not suitable to me, this should also rate it to be bad for similar users to me, this is so I am not shown results like this again as it is not for me.

- I as a user, would like to save good recommendations, this will mean more recommendations in relation to the one saved will appear, this is so I can be recommended more accurate items in the future.

DESIRABLE

- I as a user would like to view my account details, this is so I can see what is currently stored about me and whether anything needs to be changed. I should also be able to see any statistics that are related to me such as the accuracy of results for me.

- I as a user would like to change my account details, this is to cover any changes that may occur; also, being able to change the password will be a nice addition as I like to change my passwords regularly.

- I as a user would like to delete my account, this is so, if I decide I no longer want to use the software I would like all details linked to me to be deleted as I am no longer needed to be stored.

Related Functional Requirements

- *User can view account details.* **(CORE)**
- *User can change account details.* **(CORE)**
- *User can delete account.* **(DESIRABLE)**
    - *User is notified of any remaining information that may be left in the event of them deleting their account, however, any information that is not needed at all will be cleared and deleted securely.*
- User can save recommendations to a list that they can view. **(CORE)**
    - *When saying yes to the recommendation the recommendation is saved to the list of saved recommendations.*

Steps To Achieve Goal:

1. Create users has a saved recommendations page.
    a. Users can view saved recommendations in a table view
    b. Users can edit saved recommendations
        i. Either put back in circulation
        ii. Or not to show again
2. Create user account page.
    a. Users can view account details and statistics
    b. Users can edit account values such as password, email and such others
    c. Users can reclassify their account by answering questions
    d. Users can delete their account

Sprint Review

Sprint completed with relative ease, due to multiple server functions requiring to be implemented, multiple tests took up several hours of development time. All user stories relating to the general user are now complete to a good standard and user testing for user side can now be completed. From now on, all core functionality is complete along with some desirable requirements as well as the user aspect of the software solution is now complete. Overall, good development this sprint, with a minimum viable product achieved as well as designs and UX being above of what was planned, a good point has been reached.

Concerns with current timing are that certain aspects of the admin app may not be implemented due to mis-calculated timing of how long certain requirements are to implement. This concern comes from previous experience of implementing the classification algorithm taking far longer than expected and with some of the admin application aspects being that of a more advanced background, plans and thought should be put into place in the next sprints to what is truly required in the final

delivered product; if some of the additional tasks require to be cut in order to finish on time this should be done so.

Week 10: Admin Views

Start Date: 13th April - End Date: 19th April

Goal

Admins can view details about the software and how it is operating, that can see plots of user classifications and details about the user base. They can see trends in most popular items that are being recommended and the least.

User Stories To Base Development From.

DESIRABLE

- I as an admin need to be able to view user analytics such as user classification and their values, as to see how well the algorithm is clustering users together so that I can see if any users are not accurately classified.

- I as an admin would like to be able to enter data of items that the users can be recommended, this is so the users always be recommended new items.

ADDITIONAL

- I as the admin would like to view trends in the current suggestions to the users, as I would like to see what is most popular among the different groups/clusters of users. This is so I can see if the sorting algorithm is performing as expected and to also see if there is anything that the users may want more suggestions of within an area.

Related Functional Requirements

- *Admins can view user analytics.* **(DESIRABLE)**
    - *Admins can see how the users are getting classified into their clusters through a plot of the data, this way they can see if there are any outliers in addition to seeing how the clusters are classifying users.*
    - Admins can see silhouette values in addition to other K-Means values to describe how well the data is fitting to the clusters.
- *Admins can see trends in data for areas.* ***(ADDITIONAL)***
    - Admins are notified of the most popular recommendations for each user classification as well as the most popular recommendations overall.

Steps To Achieve Goal:

1. Alter admin home page/dashboard to be more like the finished idea, no need to be perfect.
2. Admin homepage displays user clustering

3. Homepage displays silhouette values for accuracy of clustering
4. Admins can see any outliers of user classification
5. Admins can see trends in recommendations

Sprint Review

This sprint has been completed, however due to the development process and choice of node package ‘skmeans’, silhouette values are not able to be displayed. This could be future work as these could be calculated manually, however due to the current time limitations this will not be the case. Development ended this week on a high, as even though not everything was implemented the usefulness of the admin page is displaying itself even with limited attainment of goals this week. Furthermore, the generated use case of admins being able to approve changes are still yet to be implemented, this was planned as extra work and would have been nice to implement if the project finished early. These ideas should be placed with future work outline.

Week 11: Admin Control of the Algorithm

Start Date: 20th April - End Date: 26th April

Goal

Admins are able to edit the algorithm slightly, in order to allow users higher accuracy in the operation of the software. they can also add new recommendations to be added to the users of the software.

User Stories To Base Development From.

DESIRABLE

- I as an admin would like to be able to enter data of items that the users can be recommended, this is so the users always be recommended new items.

ADDITIONAL

- I as the admin would like to adjust minor factors of the algorithm such as weighting of user values or the items recommendation rating for a cluster of users, this is so I can improve the accuracy of the recommendations.

Related Functional Requirements

- *Admins can enter data to be recommended.* **(DESIRABLE)**
  - Admins can manually enter new items to be recommended to the users for each area.
- *Admins can adjust algorithm if needed.* ***(ADDITIONAL)***
  - Minor changes can be made to the algorithms such as how many clustering groups and weighting of certain recommendations.
  - They can also override user classifications if they are an outlier.
  - Any other discoveries of the algorithm that may need to be monitored and adjusted.

Steps To Achieve Goal:

1. Add Page for admins to change algorithm settings
2. Research what admins could change in order to improve accuracy
3. Implement ideas found
4. Implement the editing of user values to allow the correction of outliers.
5. Add page for admins to add new movies to be added to the recommendation lists.

Sprint Review

Unfortunately, this sprint was only partly met, Admin can enter data to be recommended, however admins are unable to adjust algorithm details nor are they able to deal with users of the software. On the other hand, all core and desirable functionality is completed even with some additional functionality. This is the end of development of the applications as for adding new features. Minor edits may be made after user testing but overall progress on the application has gone well, this is due to in a matter of being a prototype, the final idea of Rico assistant can be seen and even used.

Week 12: User Testing And Final Fixes/Additions

Start Date: 27th April - End Date: 3rd May

Goal

User testing needs to run in order to test the product with the end user. Through testing we can see if user stories are met and the actual intended user is happy with the product. Evaluation of the findings will tell us of any additions/Fixes that need to occur and as these should only be minor, we can fix these in this week.

User Stories To Base Development From.

- N/A

Related Functional Requirements

- N/A

Steps To Achieve Goal:

1. Write user guides and technical guides.
2. Create testing plans
    a. Look at flow charts as an overview of the intended program flow
    b. Create a test plan to cover all aspects of the application
    c. Translate this into a format where the user test group can have a guide/path to test the software
    d. Create questions in order to evaluate the effectiveness of the HCI and application itself with the end user
3. Confirm testing group
4. Confirm testing group and plans with ethics approval
5. Complete testing with testing group
6. Analyse/Evaluate Findings
7. Fix anything that requires to be fixed
8. Any remaining time is to be used to continue development

Sprint Review

User testing couldn't have gone better, due to the care and attention that came with focusing Rico's design around UX and UI with focus to the user as well as being fully responsive, user feedback was in general very positive. This could also be due to continuous testing throughout development to allow fixes in each sprint's short comings. Further additions have been outlined from features that users would like to be added now they have viewed the finished product. Overall the feedback has been great to confirm Rico Assistants development went smoothly and its user focused design has been reached.

13.4 Project Vision

The premise of my project is an application that includes a recommendation system for sorting places to go and activities to do for bored, indecisive people. The idea is that based on statistics such as a user's age, location, transport method, budget and what sort of person they are (adventurous, outdoorsy etc.), users of similar statistics will be taken into account when calculating/recommending items to the user to base on what others enjoyed/had success doing. A user can rate these recommendations in order to improve accuracy for themselves as well as other users.

The use of an AI algorithm/Machine learning can be implemented in order to achieve this recommendation system; users will train the algorithm with their decisions; this is far better than just a set, basic recommendation algorithm but this could be used in early development. A responsive web app will be implemented as to make it accessible and adaptable to users on either a desktop or mobile device. The backend will comprise of the larger scale database that stores all the necessary information along with server-side code for all calculations for recommendations. The front-end as said will be based on a web application, this application will show activities to the user with what is best suited to them being first shown. The user can then either accept this item, save it as something they may like to do in the future, or remove this recommendation as it isn't a good match for them. The user's decision will be taken into account and if necessary, adjust the algorithm. Administration pages will be needed in order to remove/add things to do as well as manage the user base via an administration account. Later stages of development can also include the use of APIs such as to implement the use of google maps to automatically guide users to places in relation to chosen activity to do.

The most vital/important aspect of this project is the recommendation algorithm/s, not only could this algorithm be used for other data analysis, but it could also be engineered to deal with the recommendation of other items to people, such as medicines and treatments based upon what other people have had success with. This could all be put into this one web app with all different sections for what each user requires.

With life becoming more and more automated, this software can automate further aspects of life, furthermore, as with a lot of recommendations being opinion based, my software aims to give a form of statistical proof to the data being recommended.

The problem that this software aims to solve is to speed up the search for items by giving information to a user that mathematically should be of use to them. This software will also remove the need of the user performing searches and research, as all they will have to do is select the type of item they require and based upon the statistics of the user, only related information will be automatically displayed.

13.5 Functional Requirements

Main Features

- ASPECT: User Account Creation
    - *User Account Creation Page* **(CORE)**
        - *User account details are entered (Username, Name, Password, Email).*
        - *User is guided through set questions (About Me), in order to find their classification.*
        - *Details are run through classification algorithm and the user's classification ID is stored with their account.*
        - There is no link between their account and data, once data is condensed into 3 dimensions, their input data is deleted. (see user classification).
        - *User account is created and stored, user is redirected to login (to validate details) or automatically logged in?*

    - *User Classification (K-means prototype (Neural Network Later))* **(CORE)**
        - *User inputs data based on questions asked in order to locate and classify themselves to similar users.*
        - From these questions 3 values are calculated as to condense the input data into workable data. Input data is deleted/not stored only their calculated values. (See Security).
        - *These 3 calculated values are plotted using K-means clustering in order to find a user's similar user group/cluster.*
        - *User is given their cluster ID and sorted with their account details.*
        - *User can run a reclassification at any time in case of a change of interests / recommendations are not 100%.*

- ASPECT: User Login
    - *When accessing the app, if not logged in this is the page that will greet the user.* **(CORE)**
        - *If a user tries to manually access page they do not have access to they will be redirected to the last page they were on/login if they were not logged in.*

    - *User can use their login details and be redirected to the home page of the software where they can select the area that they want recommendations.* **(CORE)**
        - *Login page – this will be the home of the application if not logged in.*
        - *One user login for all users, whether admin or not (simplified process).*

        - *Any access to a page that requires you to be logged in will redirect here.*
        - *User can login using their details and be navigated into the application.*
        - *User can be navigated to the create account page if they do not have an account.*
            - *Admin accounts cannot be created, they must be created by another admin/developer.*

- ASPECT: Home Page
    - *User can navigate to available pages from here (Account, logout, help, recommendation pages, etc.).* **(CORE)**
        - *Navigation bar at the top to navigate to different pages*
        - *Navigation bar at the left in order to navigate to different recommendation areas*

    - *User can view their saved recommendations.* **(CORE)**
        - *User can view all their favourites/saved items from each area in a saved stack here.*
        - *User can remove a saved recommendation, they can choose whether to put this back into the pool of recommendations or to not be shown this again*
            - *User can see the recommendations they have chosen not to see again and add them back to the pool of recommendations or save them again. (This is like a back-up of recommendations.)*

    - *If a user needs to give additional information for a recommendation area they are queried and asked to input data on first access to the area.* **(DESIRABLE)**
        - *The user on visiting an area, such as movies, may require additional information from the user in order to categorise/classify them accurately.*
            - *When visiting the recommendation area for the first time they will be notified of this and another 'about me' section will be given to them.*

    - *User can select the area that they would like a recommendation from.* ***(ADDITIONAL)***
        - *Navigation to the area (Navigation bar?).*
        - *Each area is specific what the user wants recommendations in (Movies, Activities, Medicine).*
        - *Navigation should be seamless and easy for the user to understand.*

- *This area may have its own user classification scheme for the relating area as the base user values may not be suitable for classifying users in that area.* ***(ADDITIONAL)***
  - *This could either use the base classification values of the user on their own or with additional input from the new input data to give another 3-4 classification values in order to plot them into a specified grouping for that area.*

- ASPECT: Area specific page (area of recommendations)
  - *Recommendations are suggested to the user based on similar users interests to them.* **(CORE)**
    - *All recommendations are based on how people in the same classification of the user has responded to each recommendation.*
    - *Most suitable recommendations are shown first, then as the user goes through the stack of recommendations they will decrease in match percentage*

  - *User, through the use of the software, will give feedback on if the recommendation was good for them or not.* **(CORE)**
    - *A user saying yes to the recommendation will increase the suitability rating average for that classification/cluster group. Saying no will decrease the rating average.*
    - *How the rating is given a value needs to be calculated based upon the algorithms used. By the majority of users in a cluster preferring that recommendation a rating closer to 100 should be used; closer to 0 for when the users of that cluster have a dislike/ rejection to the recommendation.*

  - User can save recommendations to a list that they can view. **(CORE)**
    - *When saying yes to the recommendation the recommendation is saved to the list of saved recommendations.*

- ASPECT: Account Page
  - *User can view account details.* **(CORE)**

  - *User can change account details.* **(CORE)**

  - *User can delete account.* **(DESIRABLE)**
    - *User is notified of any remaining information that may be left in the event of them deleting their account, however, any information that is not needed at all will be cleared and deleted securely.*

- ASPECT: Security

- *User data is stored securely.* **(CORE)**
  - *Secure database to store all data.*
  - *No connection strings are stored/sent in plain text.*

- *Passwords are hashed and never stored/sent in plain text.* **(CORE)**
  - *Hashed password is stored, never plain text.*

- *Users personal data is taken seriously with every aim to make sure anything that has to be stored due to the nature of the software is stored securely.* **(DESIRABLE)**
  - *No direct links between the data and the user, all links if avoidable, are.*

Later Features

- ASPECT: Admin Page
  - *Admins can view user analytics.* **(DESIRABLE)**
    - *Admins can see how the users are getting classified into their clusters through a plot of the data, this way they can see if there are any outliers in addition to seeing how the clusters are classifying users.*
    - Admins can see silhouette values in addition to other K-Means values to describe how well the data is fitting to the clusters.

  - *Admins can enter data to be recommended.* **(DESIRABLE)**
    - Admins can manually enter new items to be recommended to the users for each area.

  - *Admins can see trends in data for areas.* ***(ADDITIONAL)***
    - Admins are notified of the most popular recommendations for each user classification as well as the most popular recommendations overall.

  - *Admins can adjust algorithm if needed.* ***(ADDITIONAL)***
    - Minor changes can be made to the algorithms such as how many clustering groups and weighting of certain recommendations.
    - They can also override user classifications if they are an outlier.
    - Any other discoveries of the algorithm that may need to be monitored and adjusted.

- ASPECT: Neural Network
  - *Advanced version of the recommendation algorithm.* ***(ADDITIONAL)***

- Research into using a custom Neural Network that will provide more accurate recommendations when compared to the planned K-Means algorithm.

Notable Ideas to Keep in Mind

- Clean and simple input of user data **(CORE)**
  - Users are never asked too complex of questions to answer.
  - Users data is cleansed upon input.

- Security through abstraction of the storage of user data **(DESIRABLE)**
  - *User personal data is cleared once their classification values are calculated.*

- Users must not be suggested items that are not suitable for them **(DESIRABLE)**
  - For example, users under the age of 18 shouldn't be suggested items that require/suggested for people of an age above 18.

- This application will be web based and reactive, as to make it accessible to all devices whether mobile or desktop (Many web clients too). **(DESIRABLE)**

- This will be a project made to grow (further areas for recommendations, constant checks on the effectiveness of the solution through the admin app, etc.) ***(ADDITIONAL)***

13.6 Non-Functional Requirements

Admin

- Information must be quick and easy to read and understand.
    - *The admin must be able to see how the algorithm is classifying users within a quick glance, it may not be an in-depth view but a basic overview to decide whether to investigate further.*
    - *It but be quick and easy to understand the data, as the software will operate with a large dataset it be presented in a way that everything that needs to be displayed is, and this is then displayed in a way that a user would easily be able to understand it.*
- The admin must be unable to cause a major problem in the software that would affect the operation of the application severely.
    - *Admins must be warned if any changes admins have implemented/applied that it could affect the operation of the software. Or better yet there may be a backup of before any settings admins applied, and admins shouldn't be able to change anything that would break the software.*

General User

- The user should have a suitable amount of available recommendations.
    - *The user shouldn't be just recommended one recommendation, the algorithm should allow for a suitable size of recommendations to be available at any time for ALL users.*
- The user must not be able to enter incorrect data that would affect their classification.
    - *Upon the user entering their data that allows them to be classified, this data must be cleansed, and the user must be notified of any incorrect data entered. A better solution would be, that in the vital parts of the application that take in classification data, the user must be walked through in such a manner that guides them to not enter bad data.*
- The user must find the software quick to use.
    - *The user must get the results they want in a quick way, being the main idea of the application, they don't want to be slowed down by the algorithm taking a while to classify and suggest recommendations nor do they what it be slowed down by poor navigation.*
- The user must find the software easy and intuitive to use.
    - *The software must be intuitive to use, any parts that require help or a bit more explanation will have a walkthrough to new users that help guide how to use the software. Navigation will be simple and easy to understand and only what is necessary to display to the user will be present. HCI design principles will have heavy influence on the UI design as to achieve an easy and intuitive design of the application*

13.7 Misuse Stories

Malicious Hacker

- A malicious hacker may intend to steal personal data from servers in which data is stored.
    - *to prevent this all data must be stored on a secure server with all communications of sensitive data to be in a secure encrypted format. It has been previously mentioned that any data stored that can be stored without any link to a user should be taken as an approach.*
- A malicious hacker may intend to take down the web servers through means such as DDOS/DOS attacking and other similar attacks.
    - *Protections against such attacks should be implemented, research on the top attacks for websites will be investigated to see if a protective solution can be applied to Rico Assistant.*

Underage User

- A user may intend to access parts of the application in which they are not the correct age for. For example, they may be recommended movies that they are not old enough for.
    - *Correct checks for a user's age will be made in order to prevent an underage user being suggested inappropriate material.*

Bad System Admin

- A bad system admin may attempt to edit important algorithm variables and by doing so cause the software to develop faults in the accuracy of the software recommendations.
    - *A constant back up of previous settings will always be stored along with their success rate in relations to go suggestions. Also, perhaps any changes will have to be checked and approved by another administrator.*

System Hardware Faults

- System hardware holding data may break and cause the loss of data.
    - *Back-ups of user data will be made in addition to storing data in such a way that data loss is very unlikely.*
- Systems hosting the web application may go offline and not allow access to the web applications.
    - *Well respected hosting services are to be used for hosting this web application. Research into how often failure occurs will help assist the decision on who to use. The decision to use a large-scale company for hosting is due to the affordability over creating a personalised server for Rico Assistant.*

- Systems hosting the web application may go down as important data transfers are occurring, such as a user changing their password meaning that a fault could occur causing the user being unable to login.
    - *Any changes will not be permanent until confirmation that all data needed to be transferred is complete, the user will be notified whether changed are complete or not as to not confuse the user on the status of their requests.*

13.8 User Stories

USER: Admin

CORE STORIES

- I as an admin need to be able to login, so I can fulfil my administrator duties so that I can help the users of the software.

DESIRABLE STORIES

- I as an admin need to be able to view user analytics such as user classification and their values, as to see how well the algorithm is clustering users together so that I can see if any users are not accurately classified.

- I as an admin would like to be able to enter data of items that the users can be recommended, this is so the users always be recommended new items.

ADDITIONAL STORIES

- I as the admin would like to view trends in the current suggestions to the users, as I would like to see what is most popular among the different groups/clusters of users. This is so I can see if the sorting algorithm is performing as expected and to also see if there is anything that the users may want more suggestions of within an area.

- I as the admin would like to adjust minor factors of the algorithm such as weighting of user values or the items recommendation rating for a cluster of users, this is so I can improve the accuracy of the recommendations.

USER: General User

CORE STORIES

- I as a user of the software want to be able to create an account, this is so I can access the software and be able to be recommended items such as movies to watch.

- I as a user of the software want to be able to be classified into a group of similar users so that I can be recommended items that are popular and well suited to those like me.

- I as the user of the software, would like to log into my account and be able to access my recommendations, saved or new.

- I as a user of the software, would like navigation of the application to be easy and clean, automatic redirects to where I need to be would assist in this. This is so I am always at the right place at the right time.

- I as the user would like to be able to navigate to all available pages with ease, this is so I can access all items that I want from the software, such as my account details and my saved recommendations.

- I as a user, would like to view my saved recommendations that I have selected to save, this is to save me time from getting my favourite/most suitable items re-recommended through the recommendation algorithm.

- I as the user would like to remove recommendations from my saved stack, I can select whether they are put back ready to be re-recommended or to not be presented with them again. This is so I can manage my list of saved recommendations.

- I as a user, would like to be suggested recommendations for my chosen area that are most suitable for myself and users similar to me, this is because this is the main idea of using this software.

- I as a user would like to be able to reject recommendations to not be shown again as they are not suitable to me, this should also rate it to be bad for similar users to me, this is so I am not shown results like this again as it is not for me.

- I as a user, would like to save good recommendations, this will mean more recommendations in relation to the one saved will appear, this is so I can be recommended more accurate items in the future.

DESIRABLE STORIES

- I as a user would like to say that a recommendation is good, but not for me right now, this will increase the chance of this item popping up again, but it will not be saved.

- I as a user would like to view my account details, this is so I can see what is currently stored about me and whether anything needs to be changed. I should also be able to see any statistics that are related to me such as the accuracy of results for me.

- I as a user would like to change my account details, this is to cover any changes that may occur; also, being able to change the password will be a nice addition as I like to change my passwords regularly.

- I as a user would like to delete my account, this is so, if I decide I no longer want to use the software I would like all details linked to me to be deleted as I am no longer needed to be stored.

- I as a user would like my details to be sorted securely and security over my personal details to be taken seriously, this is so data breaches of my data does not occur.

ADDITIONAL STORIES

- I as the user of the software would like to select an area of recommendations, this is so I can be recommended items closer to what I need, e.g. a movie to watch, or an activity to do. (this is all just adding more areas to be recommended)

13.9 Risk Assessment Plan

Risk Identification	Risk Monitoring	Risk Treatment
Storage of user's data GDPR - The storage of user's data must be secure and abide by GDPR laws. **Severity 8	When coding the project precautions and monitoring of what data will be stored must be taken into account. To reduce the impact a log of what needs to be stored securely must be noted and correct security implementations must be included within the project.	Data breaches must be prevented by implementing the correct and appropriate security. If a data breach occurs, then this must be reported to the project manager who will take the appropriate measures to rectify the problem.
Recommendations are not 100% safe for the end user **Severity 8	Investigations to each area of application recommendations and their potential safety hazards in relation to a user must be noted and considered of their impacts. Appropriate warnings for the end user must be implemented into the development of the project in order to be certain that the user will be warned and notified.	User needs to be aware that the recommendations to them could pose a danger to themselves, an example being a recommendation to a theme park to someone with an underlying heart problem. To prevent this a user needed to be notified on signing up to use the software that not all recommendations may be safe for them to do so. Each activity could have its risks outlined and categorised.
Not meeting end user requirements **Severity 7	Without correct identification of what the end user will require, or how the problem of the project will be solved, the project will not fulfil its purpose. By generating user stories and a thorough functional specification, it can be certain that what is required will be met (given enough time). Also, by monitoring throughout development whether these have been met we can apply sprints to ensure that with the timescale given, all end user requirements will be met.	In order to correct any mistakes user testing along with usability testing is to take place. From these a guided tour covering all aspects of the intended application will be addressed and tested, this will be in addition to directed questions if their requirements have indeed been achieved.

Loss of work/data **Severity 4	Loss of work by hardware failure or loss of storage device is a prominent problem when the correct precautions aren't taken. Although preventing loss of data is vital it is very easy to prevent. The use of an online back-up of the files along with a version control system (such as GitHub), will mean that not only will all files be backed up offsite, if a current version of software breaks the project it is easy to revert back before the changes.	As mentioned before, the use of an online back-up/version control system will allow the restoration of any loss of work. It must be made clear to any developer of the project that all work must be backed up and committed to the GitHub repository.
Time scale and scope **Severity 6	Making sure that the project can be developed inside the timescale given is important as well as making sure that the scope of the project is suitable for the development life timeframe. As this is a project that arguably has no end and constant development can be made, the only worry is that the main ideas of the vision can't be implemented within the timeframe available. To prevent failure of getting the necessary functionality complete an agile approach to project development as well as correct time management must be applied. Constant monitoring of the success of these plans are also vital, as plans alone will not lead to actually sticking to them.	In the event that timescale or scope needs to be adjusted due to discoveries throughout development allocated contingency time will be used in order to get back on track. If in the event of contingency time not being enough or not being available, then a meeting with the project lead and project supervisor must take place, here the focuses of development can be refocused into the most necessary first.
Problems in development causing slowing of development	Problems may arise in development that cause slowing of development will be prevalent, although certain in	In the event of delays due to problems arising in the development cycle causing a week's sprint to extend

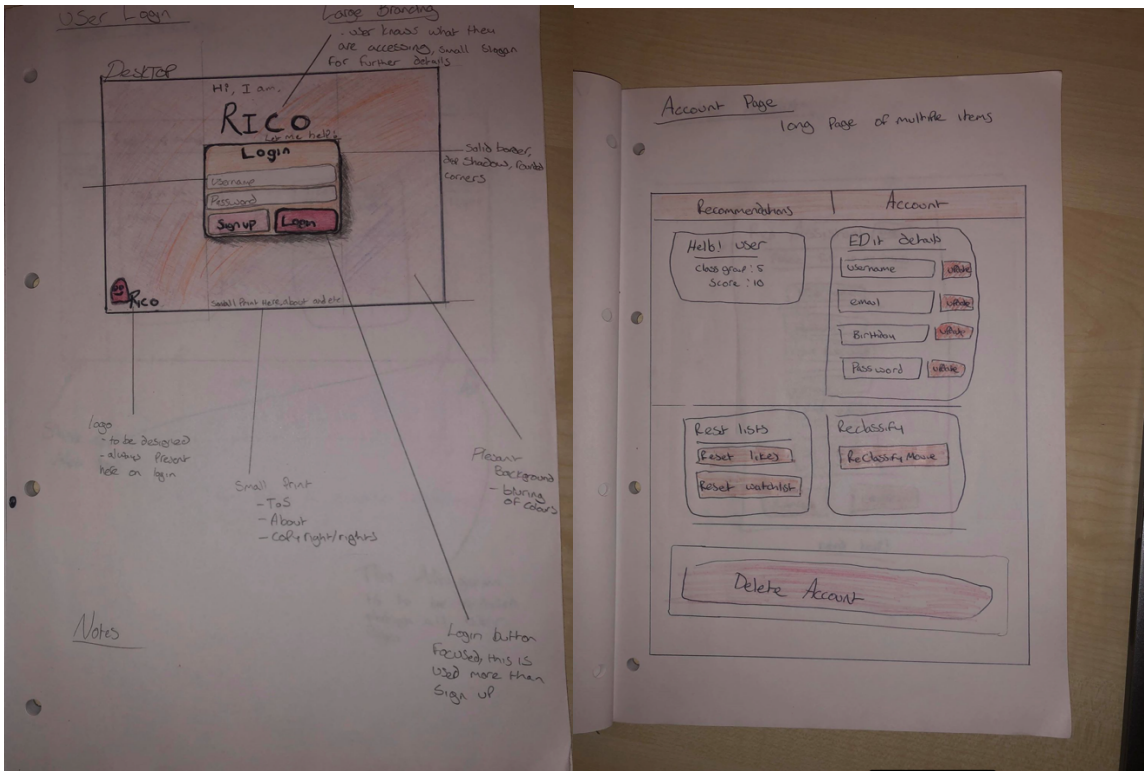
**Severity 6	any programming lifecycle that this will occur again with correct planning of contingency time will help give notice to these areas before they arise. Agile approaches to development help to aid and monitor each week's progress in order to log if the slowing of development is occurring.	into the next, planned contingency time will be used. Contact with supervisors and getting help as soon as possible will clear up any problems as quickly as possible, also with agile development while waiting for any help needed something else can always be in development.
Overworking/stress **Severity 4	Overworking will lead to stress, and this will lead to the slowing of development. Constant reviews of wellbeing and dedicating a balance to work and relax time will prevent any overworking. By planning to finish core functionality development early should mean the avoidance of a feeling of rushing towards the end of the project due to time restraints.	If overworking causes stress or vice versa immediate contact to the project supervisor must take place. Arrangements for time to relax and an adjusted schedule must be constructed and confirmed in order to get back on track as soon as possible. Wellbeing must be taken seriously and check-up's after must take place also as once it has occurred once, the likelihood of overworking and stress to re-present itself is high.
Physical strain **Severity 2	Repetitive Strain Injury and physical strain on the body (back, neck, wrists etc.) in this work place setting is a reality. Correct work place ergonomics must be implemented where possible along with regular breaks in order to prevent injury and strains as previously mentioned. To also prevent eye strain glasses are to be worn if prescribed at all times. Constant breaks from sitting at a desk are to be mandatory to not only stretch the body but to relax the eyes and mind.	If in the event of physical strain of any team member of the project occurs, then a self-assessment will take place. If deemed necessary, rest will follow and if in the case there is no improvement of symptoms medical help from a professional must be requested. This must all be logged with the project supervisor and project lead.

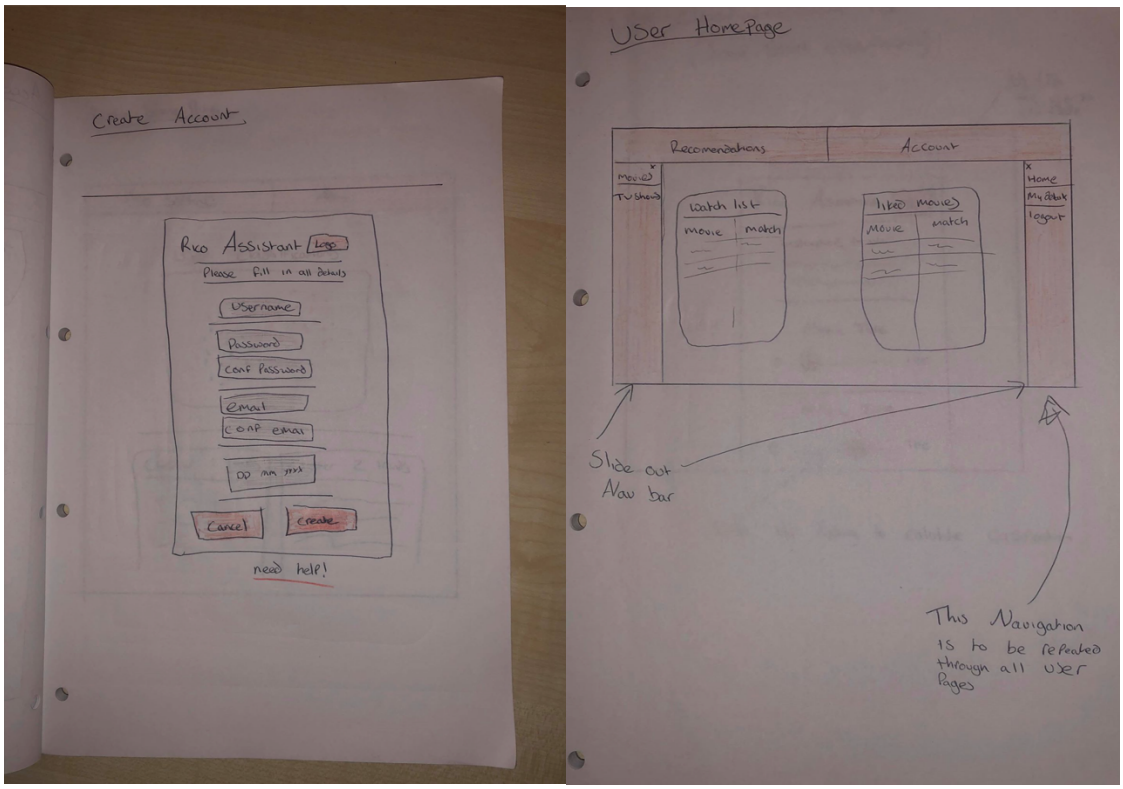
Incompatible components/ technologies **Severity 5	If within development, it is discovered that components of the project are incompatible or certain coding languages cannot do the functions required from the project, this could cause development to come to a complete stop while a work-around is found. With proper planning and research in to what components and technologies are needed along with their compatibilities before coding development, this issue should never arise, as long as the components technologies do not change their compatibility. However, with regular monitoring of the component and technologies used and with their latest compatibilities checked, awareness of any changes will be made as soon as possible, and a solution can be found.	In the case of incompatible technologies but a solution can be found contingency time must be used in order to fix the problem at hand. If a solution is not found then a customised implementation can be created, an example being if a node package is not available then a custom new package could be developed and implemented. If neither of these fit into the timescale available, then this must be noted as future work and either a different approach of development must be considered or if the certain aspect can be cut then the most appropriate action must take place.
Software may not work on all devices **Severity 5	Being a web-based app, many technologies will be used whether that be a physical device change (desktop/mobile) or a different browser being used. Research into the most used browsers and compatibility with functions used in the code can be checked to see if any special requirements in the code is needed. As for the server side it must be checked that any backend technologies are suitable for where they will be deployed along with making sure the server itself will be able to deal with the user base	The likelihood of this occurring although high it is one risk/problem all software will face. It is near impossible to get software working for all devices due to their vast configurations. However the most used browsers will be researched and the software solution will aim to support the most used. If in the case problems are present on certain configurations this must be noted and put forward to future work/bugs. The user must be able to view the supported

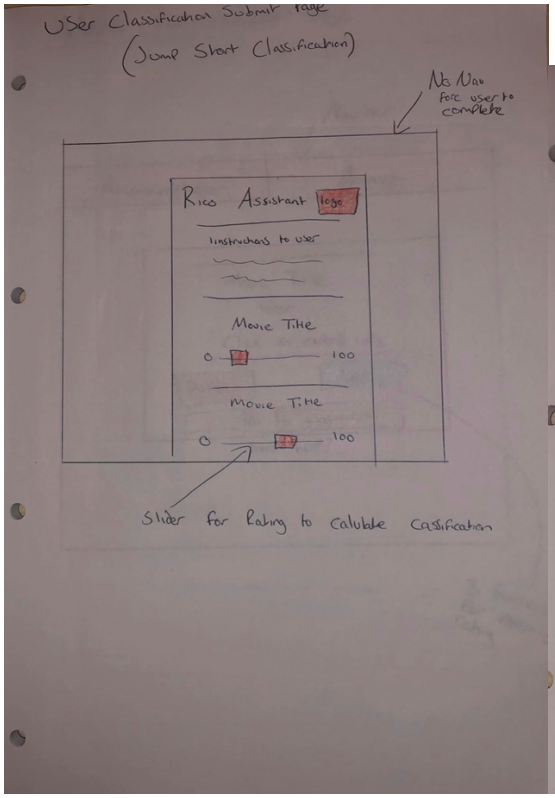
	whether that be the number of users themselves or how they will be accessing the server.	configurations and browsers.
Risk Identification	**Risk Monitoring**	**Risk Treatment**

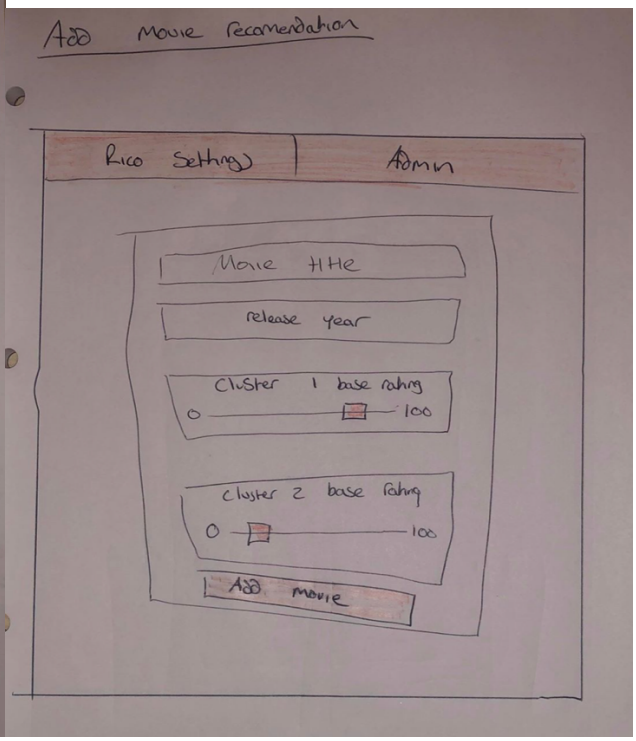
*** Severity rating on a scale of 1-10 with 10 being most severe (Must be monitored) and 1 being of little concern.*

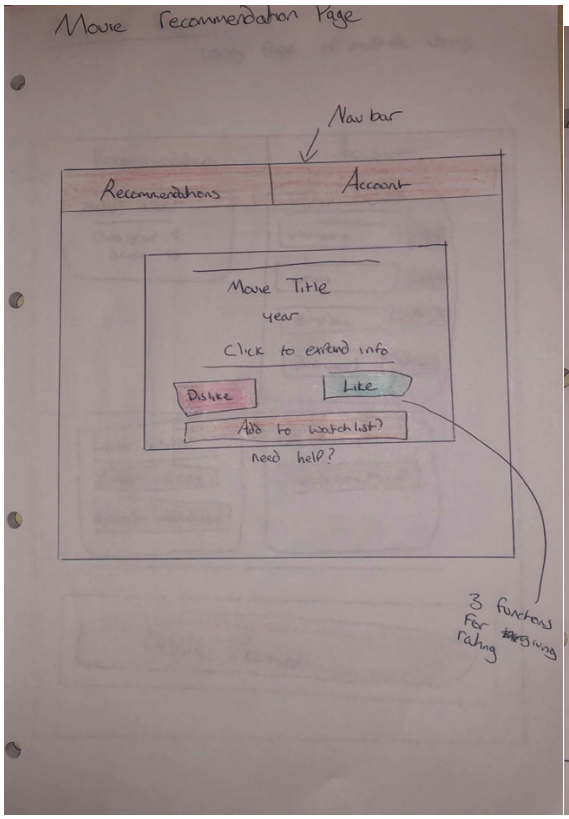
13.10 User Interface Designs

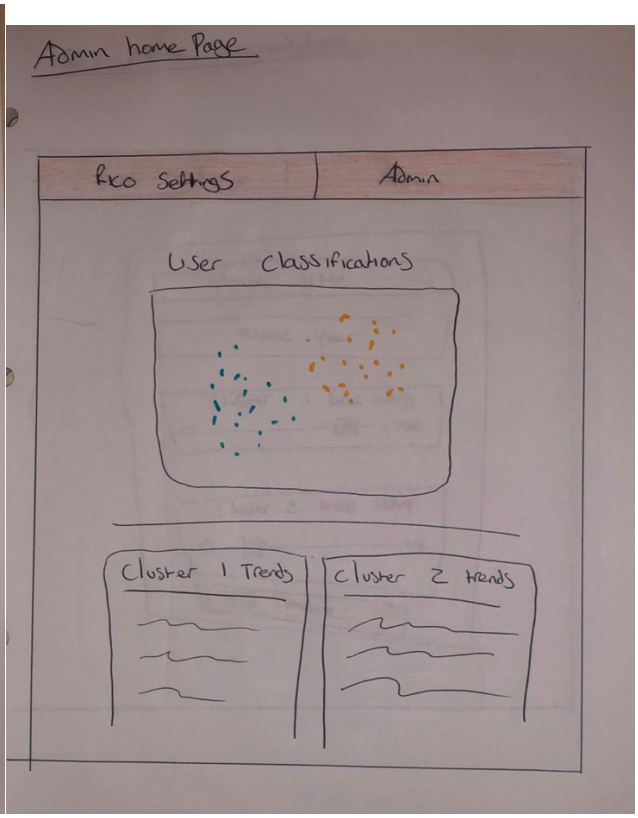





User Classification Submit Page
(Jump Start Classification)
No Nav
force user to
complete
Rico Assistant
Instructions to user
Movie Title
0
100
Movie Title
0
100
Slider for Rating to calculate Classification


Add Movie Recomendation
Rico Settings
Admin
Movie title
release year
Cluster 1 base rating
0
100
Cluster 2 base rating
0
100
Add movie


Movie Recommendation Page
Nav bar
Recommendations
Account
Movie Title
year
Click to extend info
Dislike
Like
Add to watchlist
Need help?
3 functions
For
rating


Admin home Page
Rico Settings
Admin
User Classifications
Cluster 1 Trends
Cluster 2 trends

13.11 Functional Test Plan and Results

Requirement	Checks	Data	Actual Result
User Account Creation Page	Check client-side validations only allows valid data Check a user can create an account Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	Username: bob Password: bobpassword Email: bob@rico.com birthday: 24th July 1997	As Expected
User Classification	Check that a user is classified with expected values and classification group Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	Username: user Password: password	As Expected
User Login	Check that validation of inputs is present Check that a user can login Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	Username: bob Password: bobpassword	As Expected
View Saved Recommendations	User homepage should display watchlist and likes movies User should be able to edit these lists Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	Username: bob Password: bobpassword	As Expected

Navigation of pages	User should be able to navigate to all pages with an active marker displayed for each page Navigation should be simple and easy with the sliding bars each side of the page Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	Username: bob Password: bobpassword	As Expected
Movie Recommendation	A user should be given recommendations based on classification group User can like and rate, dislike and add a movie recommendation to their watchlist When all recommendations are given, a message is displayed to a user Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	Username: bob Password: bobpassword	As Expected
User Can Change Account Details	A user can see account details User can edit account details and they are updated on the database User can reset their classification groups User can reset their interaction lists with movies. Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	Username: bob Password: bobpassword	As Expected
User Can Delete Account	User can delete their account	Username: bob Password:	As Expected

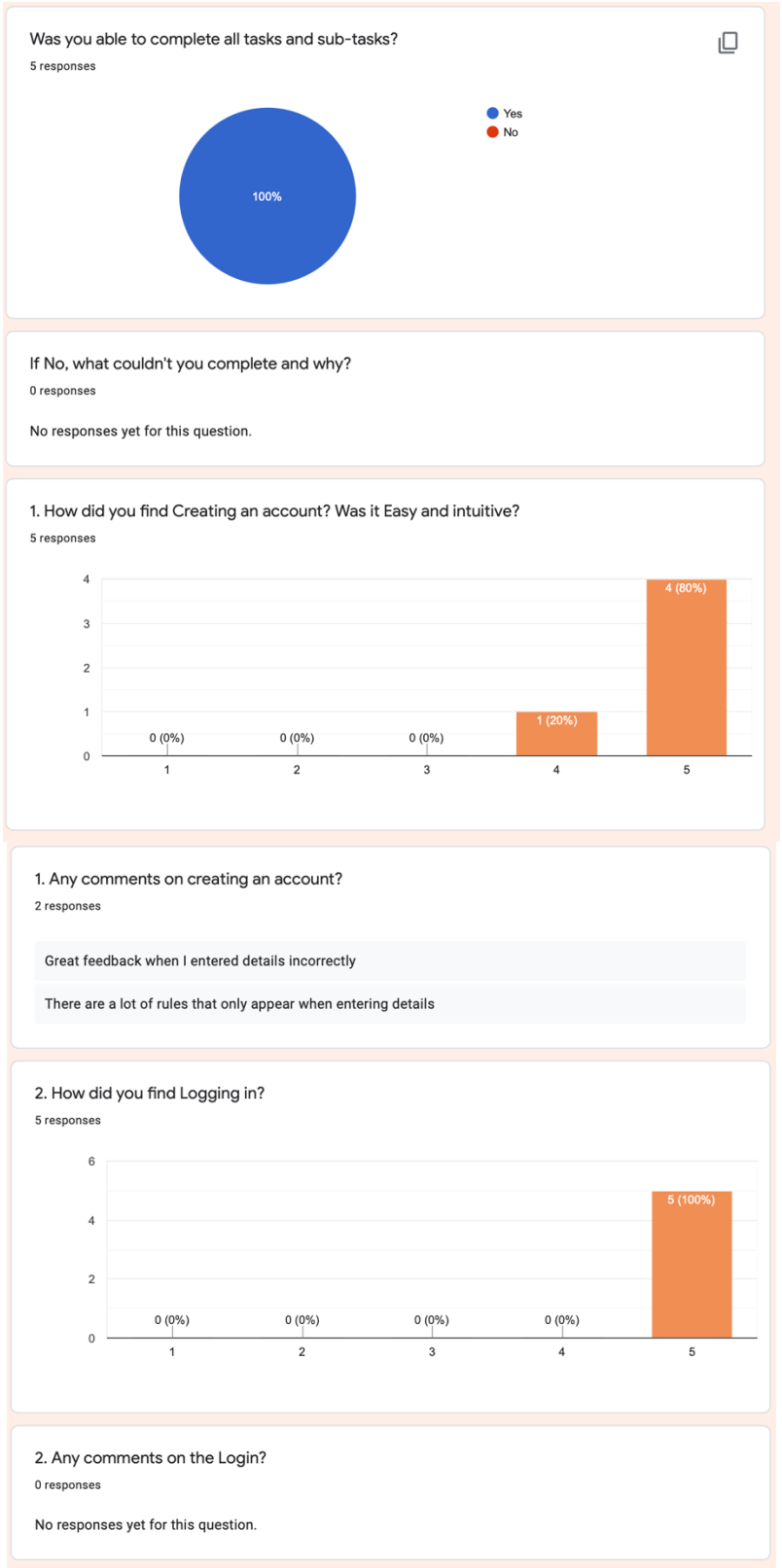
	All details are deleted permanently Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	bobpassword	
Admin Can Login	Admin can log in and be taken to their admin home page Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	Username: admin Password: password	As Expected
Admin Can See Classification Algorithm Performance	On Admin login, a clustering scatter graph of users is displayed on the dashboard Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	Username: admin Password: password	As Expected
Admin can see recommendation Trends	Top movies for each cluster are displayed on the Admin dashboard Check all unit tests are passed Check that error messages are displayed to user Check that validation messages are displayed to the user	Username: admin Password: password	As Expected
Admin can add new recommendations	Admin can add a new movie to the recommendations The added movie is able to be displayed Validations of inputs are present and functional Check all unit tests are passed	Username: admin Password: password Movie Title: Avatar Year: 2010 Ratings all 100	As Expected

	Check that error messages are displayed to user Check that validation messages are displayed to the user		
Admin can Adjust algorithm	Not implemented	Not implemented	Not implemented

13.12 User Testing Results

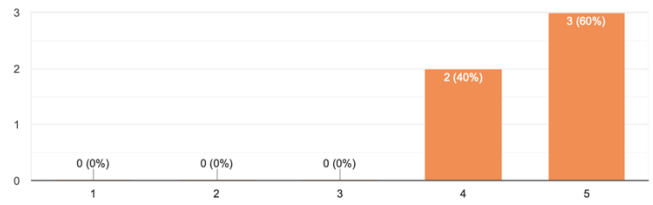
General User Test Results

Question form: https://forms.gle/DwwBGzv71GFdtGrc7



3. How was Navigating to Movie Recommendations?

5 responses



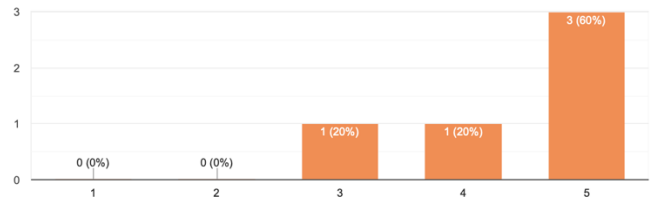
3. Any Comments?

1 response

The questions is not very clear, Navigation is simple once I know of the application. but the question confused me.

4. How easy was getting classified for movies?

5 responses



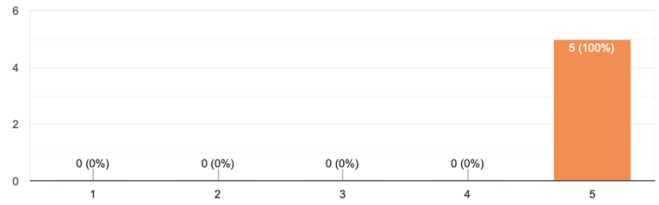
4. Any comments on Movie Classification

1 response

Perhaps too many questions, Does this only happen once?

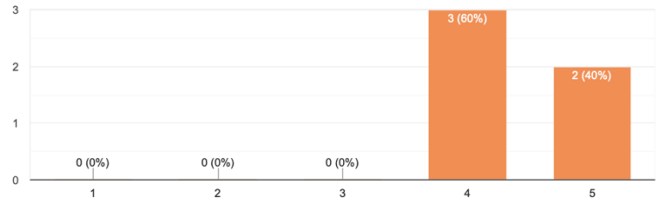
5. How easy was getting Movie recommendations?

5 responses



5. Where the recommendations of interest to you, or in other words were movies of interest displayed to you?

5 responses



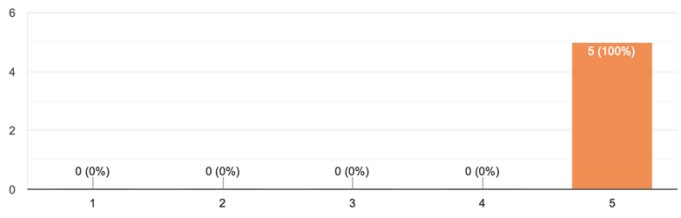
5. Any comments on the recommendations?

0 responses

No responses yet for this question.

6. How easy was editing your watchlist and liked movies?

5 responses



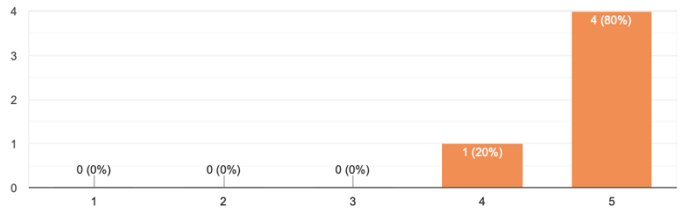
6. Any comments on your saved lists?

0 responses

No responses yet for this question.

7. How easy was editing account details?

5 responses



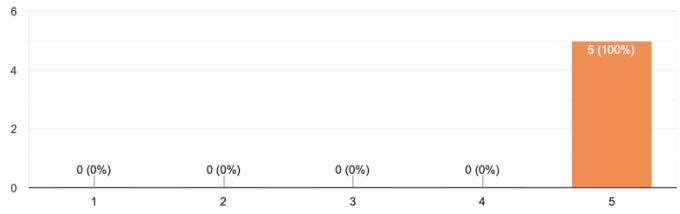
7. Any comments for editing account details?

1 response

There is a lot of options on this page

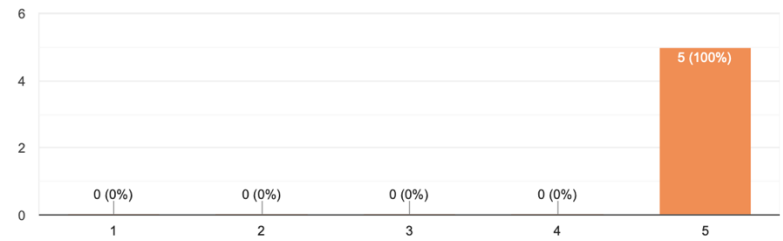
8. How easy was logging out?

5 responses



8. Does the application adapt to mobile?

5 responses



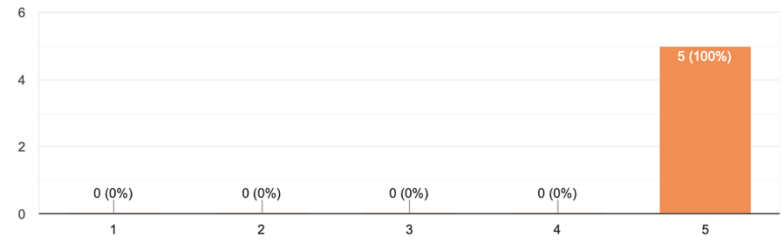
8. Any comments of logout and mobile views

1 response

Mobile view is just as good as desktop

9. How easy was deleting your account?

5 responses



9. Any comments of account deletion?

0 responses

No responses yet for this question.

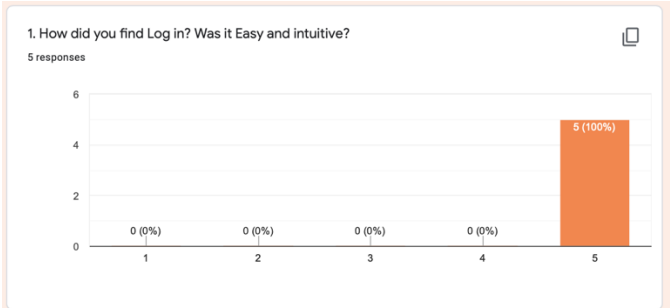
Any overall feedback for the Application?

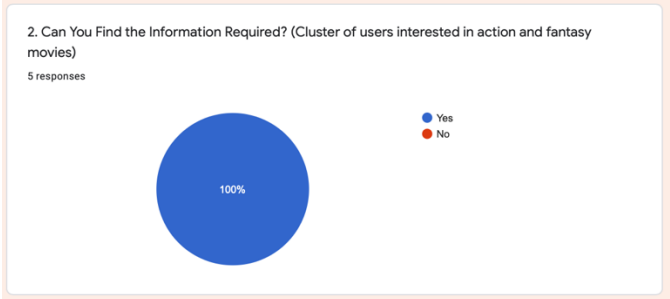
1 response

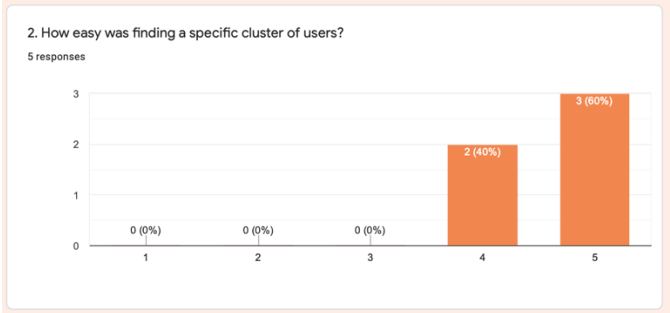
Very professional and finished, does not seem like a unfinished product. feels complete

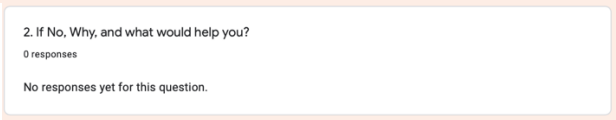
Admin User Test Results

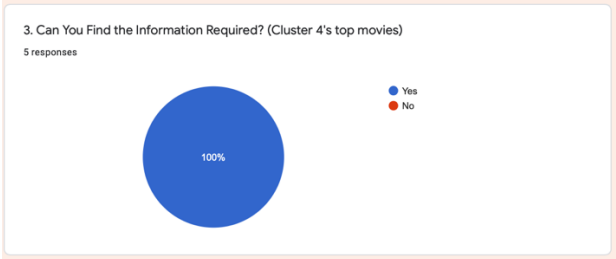
Question form: https://forms.gle/3NEYFFuhmhwEtRBJ7

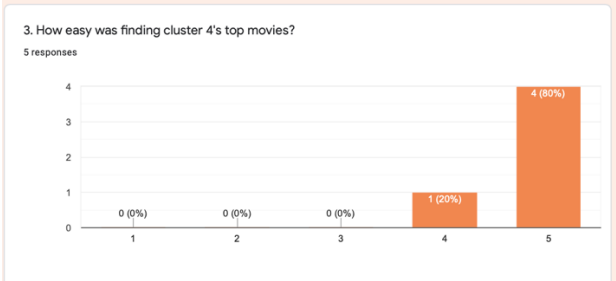


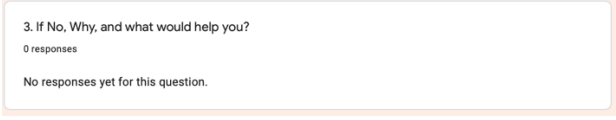






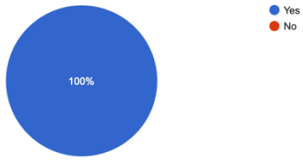






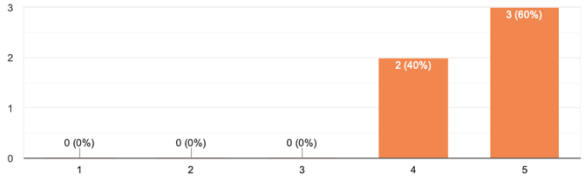
4. Can You Find the page Specified (Add Movie)?

5 responses



4. How easy was navigating to add movies recommendations?

5 responses



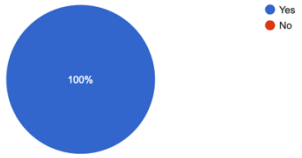
4. If No, Why, and what would help you?

0 responses

No responses yet for this question.

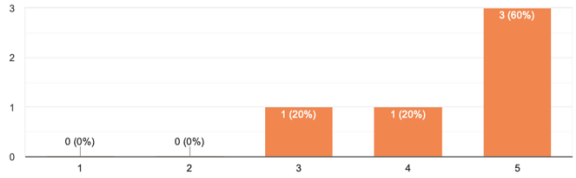
5. Can You add the suggested Movie?

5 responses



5. How easy was Adding a new movie recommendation?

5 responses



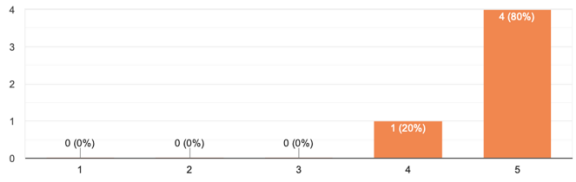
5. If No, Why, and what would help you?

0 responses

No responses yet for this question.

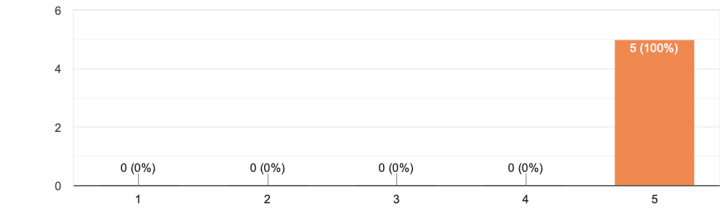
6. How did you find Logging out? Was it Easy and intuitive?

5 responses



7. Do you feel the application scales well on mobile?

5 responses



Any overall feedback for the Admin Application?

3 responses

Great messages/alerts to confirm actions

I like the error checking you have in place, it let me know I had an incorrect password easily

I wasn't too sure on what a base rating should be, I entered what I felt was correct based on instructions

13.13 Application Icons and Branding