COMP208

Group Software Project

Design Review Document

Daniel Cox
Thomas French
Lucy Rogers
Kristian Rowan
Eklavya Sarkar

17th March 2017

University of Liverpool

1. Summary of Proposal

1.1 Background, Aims and Objectives

"Scran is a user-oriented application that aids in the decision making process when choosing a restaurant to eat at, and more specifically a dish to eat at that restaurant. *Scran* will maintain, search and track user and restaurant data to help its users to choose the dish they didn't know they wanted.*"*

Scran is an app designed to aid in decision making when deciding a restaurant to go to, and more specifically a dish to eat at that restaurant. The main idea for the app has evolved during the project so far, however we have a list of essential features it needs to include as well as some desirable features and potential extensions.

Essential features:

- *Scran* will search for restaurants contained in the database in 3 different ways: manually entering the restaurant's name in the search bar, filtering restaurants according to criteria or using the GPS functionality to find restaurants near you.
- *Scran* will filter the dishes from the restaurant according to: type of dish (starter, main dessert etc), contents of dish (including dietary constraints), spice rating, previous user ratings, suggested dishes for that user (based on previous preferences, if any have been given).
- *Scran* will give the detailed nutritional information for each dish listed. This will include total calories, fat content, saturated fat, salt, sodium, carbohydrates etc.
- *Scran* will allow users to rate each dish using a traffic light system - green for "love it", yellow for "like it" and red for "dislike it". These ratings will be used to filter the user's future preferences.
- *Scran* will have 2 classes of user - free user and paid member (who pay a one-off fee). A free user has access to the basic version of the app but a paid member has the same access as free users as well as extended features, such as leaving comments on dishes. The full list of extended features available to paid members are given below.
- *Scran* will have administrators who will approve the comments left by members (to ensure appropriate language used etc), accept new restaurants and memberships and edit information for a dish/restaurant. They are also able to block/ban users if required, and have access to information not visible to normal users on the app.

- *Scran* will be connected to a database which will store all of the relevant information regarding dishes, members, restaurants etc.

Desirable features:

- *Scran* will be extended to also recommend takeaways as well as restaurants. This could then be linked up with 'Deliveroo' or 'Just Eat' so that when a takeaway dish is recommended to a user they will be able to order the item via the relevant app.
- *Scran* will include a "star rating system" for each restaurant that will be based on the reviews left by users. If a restaurant gets a lot of users selecting 'green' for a certain threshold of their dishes, they will earn a star. The more stars a restaurant earns the more they will be featured and recommended to other users.

1.2 Necessary Changes to Specification Based on New Information/Understanding

- Each user of the app will make an account, regardless of whether they intend to be a free user or paid member. This will make it easier to store user information and update their preferences on the database.
- We will not be considering the "offline mode" as it will make the app too 'bloated' for the user.
- The user will be able to change the language of the app by selecting a button at the top. For the purposes of the project this will be done using 'Google Translate' but we would consider asking native/fluent speakers to translate the words after the main project is completed. This addition was suggested to us by our reviewer for the requirements section (Sebastian Coope).

1.3 Summary of Research & Analysis

Since the completion of the requirements stage of our project, the group has (both collectively and individually) considered new and improved ideas based on analysis of our work so far, as well as researching other programs and apps (that being, programs that allow more efficient design and implementation of our ideas, and apps that work in a similar fashion to the one we plan to create). This has culminated in new ideas and plans for our project to be used in both the design and implementation stage, as well as additions to the services provided by our app.

After looking at the design of our system, we noticed that we had not made plans for any kind of application system for our "free users". Due to the lack of personal identification made by these free users in our initial design plans, it would be very difficult to effectively record data that they provided without risk of incomplete data and repeated data in our database. After some discussion we decided that since

it was highly likely that a large number of our user base would be these “free users” we would implement an application system for the “free users” and so included these plans in this design document.

During our requirements analysis review, the reviewer for our group (Sebastian Coope) made a couple of suggestions to us to improve the efficiency of the implementation of our project, the most promising of these ideas, we felt, was the use of the PhoneGap app which would help us to bring our app to mobile devices, both android and iOS. After researching PhoneGap and considering the support it would offer us, we decided we would use PhoneGap in order to take advantage of the ability to create an android and iOS app quicker so we could dedicate more time to effectively implementing the services we want our system to provide, as well as refining the system to make it better.

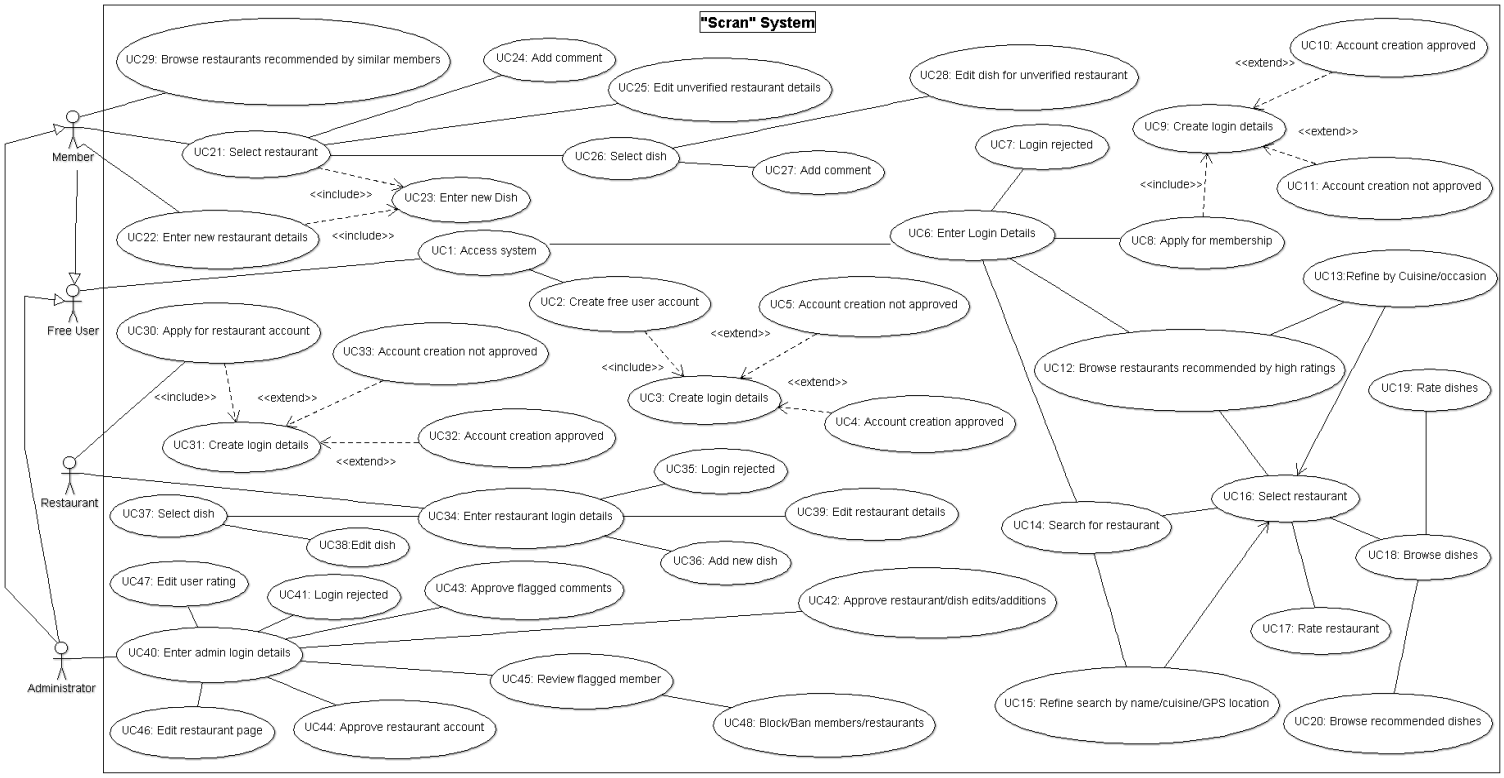
Finally after analysing the services our group had initially decided our app would provide, we decided that the (paid) members were not being offered enough services/incentives for their membership. Since then we have researched other apps and brainstormed a few ideas in order to solve this issue and decided that our app would offer exclusive deals and offers with participating restaurants (with their permission) as well as offering exclusive access to the services we provide concerning takeaways. These exclusive services will largely be the same as those services for restaurants only they will of course concern takeaways instead.

2. Design

2.1 System Architecture

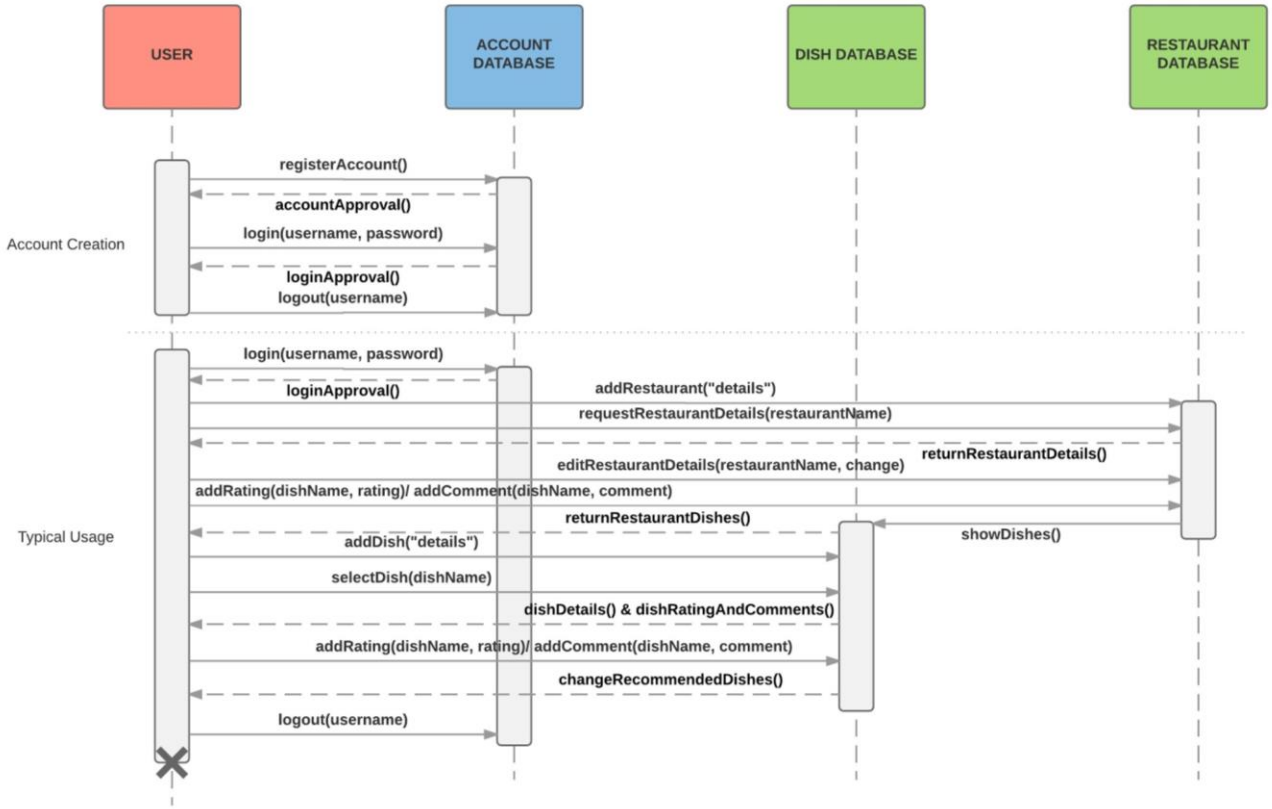
Use case descriptions for the above use case diagram can be found in the appendix.

A larger version of the Use Case Diagram is also in the Appendix



2.2 System Design & Description

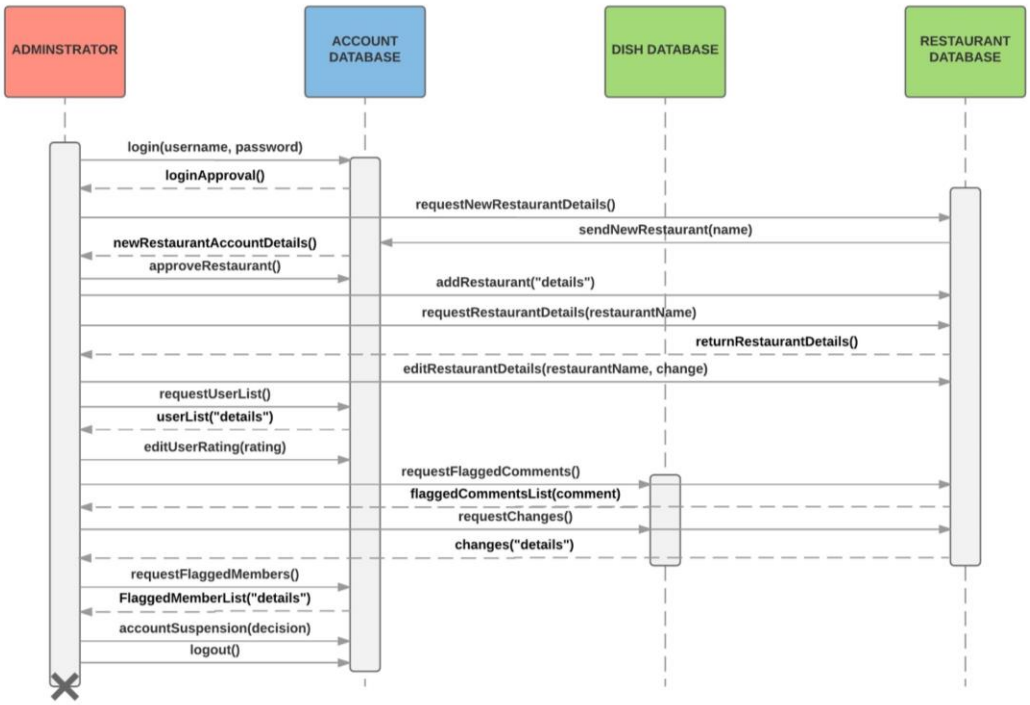
2.2.1 Sequence Diagram



This sequence diagram shows the interaction between users and the various databases. This sequence diagram has very common features across the three different users: Free User, Member and Restaurant. The Member (a premium user) has access to all of the methods in the diagram, whereas Free Users lack the ability to *addRestaurant, editRestaurantDetails, addComment* to restaurant Database*, addDish, addComment* to dish Database.

The Restaurant *accountApproval* method is different to the free user and member approval of confirming email address by clicking the link in the "Account Creation Confirmation Email", instead, once the restaurant is registered, an Administrator will contact the Restaurant to verify their identity. In addition, Restaurant users will not be able to access the following methods: *addRating* and *addComment* to dish Database and restaurant Database.

A solid line indicates the object sending a request whereas the dotted line indicates a response based on the request. Where the parameter of the method is "details", it is constructed of various different variables.

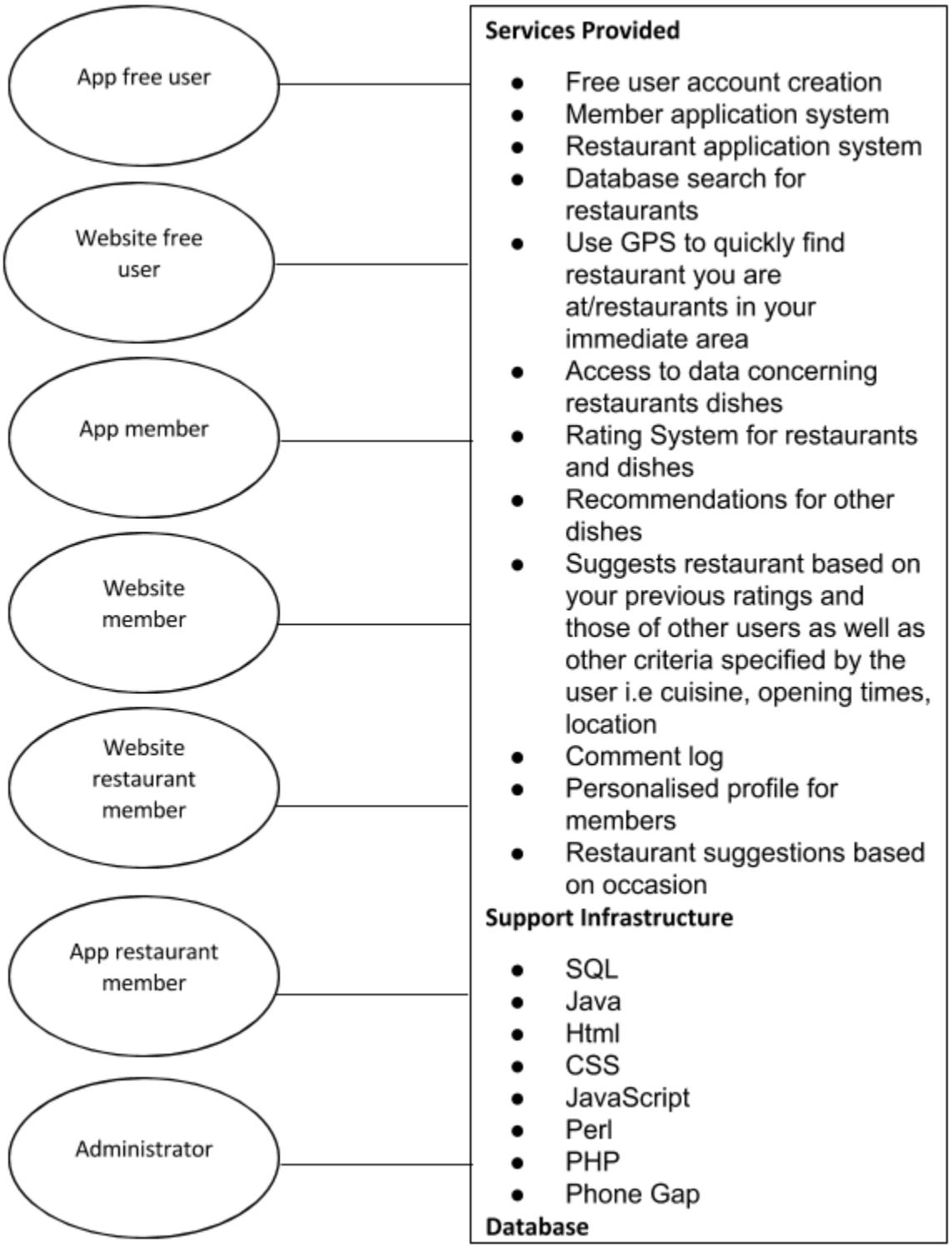


The Administrator user is able to act as a normal Free User and a Member - it inherits both of those classes so can use all the methods from the first diagram. In addition to those methods, administrators can do these methods too. Administrators will access these methods through a different portal to the standard *Scran*, this portal will be in the style of a control panel.

2.2.2 Data Dictionary

We have maintained a data dictionary throughout the process of the project so far, to ensure terms are fully understood by all team members and are referred to in the correct context. The full data dictionary for our project can be found in the appendix.

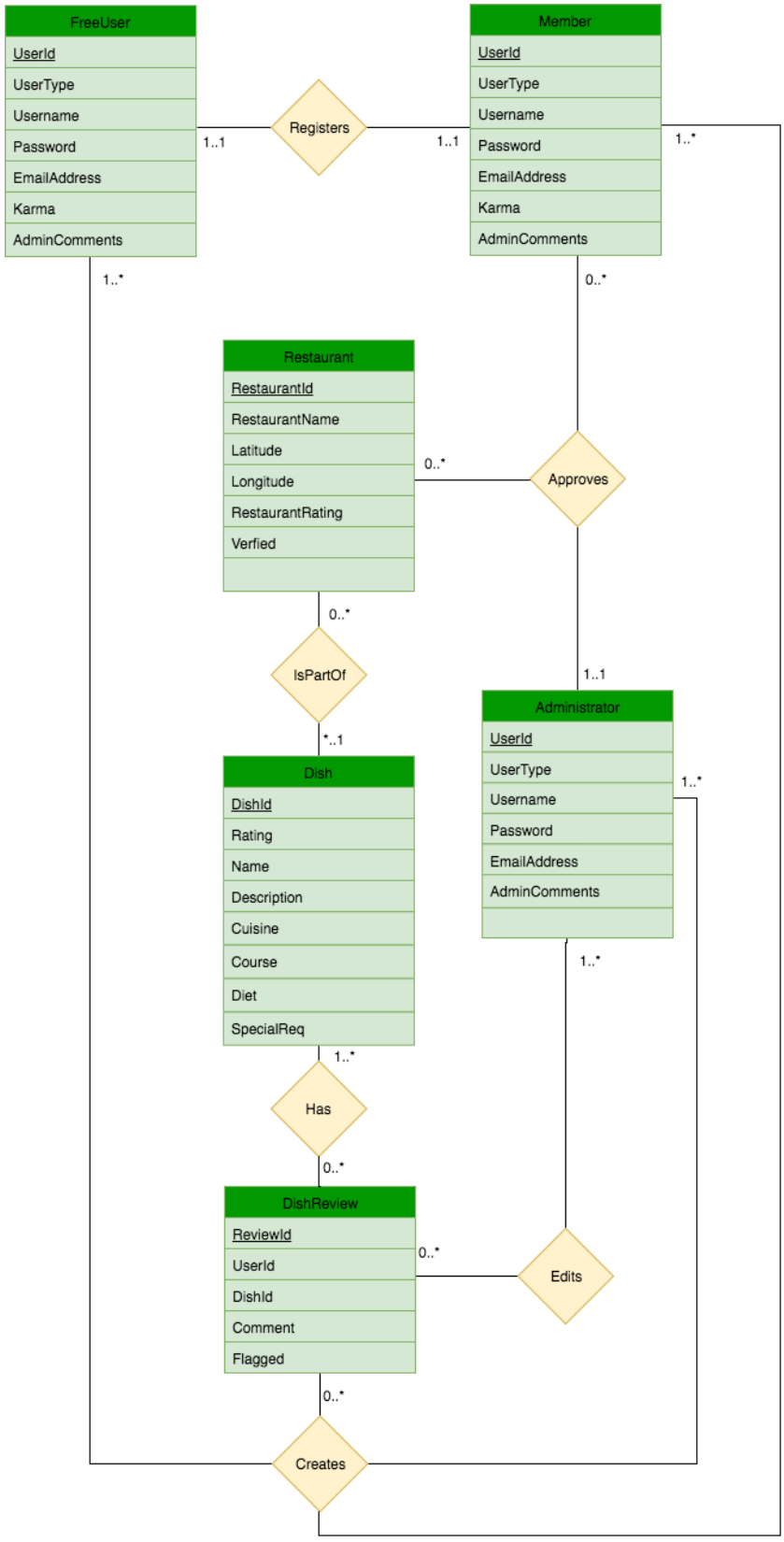
2.2.3 System Boundary Diagram



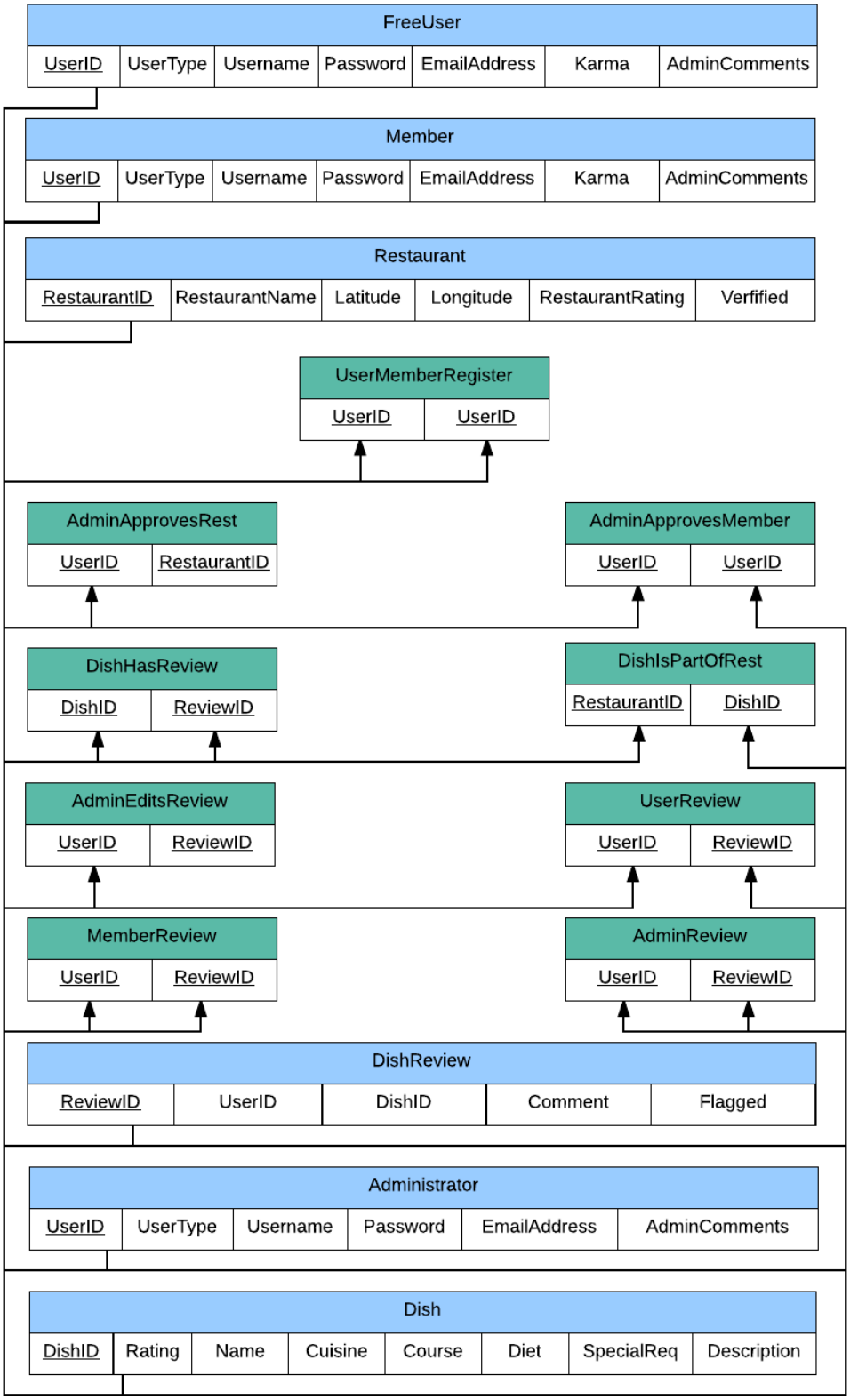
2.2.4 Table Structures

1. **Entities**
    a. FreeUser
    b. Member
    c. Restaurant
    d. Administrator
    e. Dish
    f. DishReview
2. **Relationships**
    a. Free user *registers* (to become) Member
    b. Restaurant *has* Dish
    c. Dish *IsPartOf* Restaurant
    d. FreeUser, Member, Administrator *creates* DishReview
    e. DishReview *IsPartOf* Dish
    f. Administrator *edits* DishReview
    g. Administrator *approves* Restaurant, DishReview, Member

ER Diagram


FreeUser
UserId
UserType
Username
Password
EmailAddress
Karma
AdminComments
Registers
1..1
1..1
Member
UserId
UserType
Username
Password
EmailAddress
Karma
AdminComments
1..*
1..*
0..*
Restaurant
RestaurantId
RestaurantName
Latitude
Longitude
RestaurantRating
Verfied
0..*
Approves
0..*
IsPartOf
*..1
1..1
Administrator
UserId
UserType
Username
Password
EmailAddress
AdminComments
1..*
Dish
DishId
Rating
Name
Description
Cuisine
Course
Diet
SpecialReq
1..*
Has
0..*
1..*
DishReview
ReviewId
UserId
DishId
Comment
Flagged
0..*
Edits
0..*
Creates

Logical Design



2.2.5 Physical Design

We would be using MySQLWorkbench to implement the tables in the database using MySQL and PHP to interact from the app or website to the database. To test the tables, we would create a .csv file with sample values for restaurants and users, and import it on the DBMS to sync and populate the system.

The primary keys would be as indicated (underlined in each table) above in the logical design, and the foreign keys are represented in the relationship tables, which are there to optimise performances.

The design is atomic, as all fields contain only a single value fields and in all tables, columns are unrelated, so there are no repeating groups, therefore the design is in at least 1NF. All entity table fields are only defined by primary keys, therefore the tables are in 2NF.We can also see that there are no transitive dependencies, therefore it is also in 3NF.

The data types for IDs will be ints, and most of the other attributes will be text-based, and therefore in varchar, and a couple in boolean form.

All the attributes of each entity is shown in Entity-Relation diagram, and those will be carried over in the actual SQL implementation of the tables.

Free User

Column	Datatype	Notes
UserID	int(10)	Primary Key
UserType	varchar(40)	
Username	varchar(40)	
Password	varchar(40)	
EmailAddress	varchar(40)	
Karma	int(4)	
AdminComments	varchar(40)	

Member

Column	Datatype	Notes
UserID	int(10)	Primary Key
UserType	varchar(40)	

Username	varchar(40)	
Password	varchar(40)	
EmailAddress	varchar(40)	
Karma	int(4)	
AdminComments	varchar(40)	

Restaurant

Column	Datatype	Notes
RestaurantID	int(10)	Primary Key
RestaurantName	varchar(40)	
Latitude	varchar(40)	
Longitude	varchar(40)	
RestaurantRating	int(1)	
Verified	boolean	

Dish Review

Column	Datatype	Notes
ReviewID	int(10)	Primary Key
UserID	int(10)	Foreign Key
DishID	int(10)	Foreign Key
Comment	varchar(250)	
Flagged	boolean	

Administrator

Column	Datatype	Notes
UserID	int(10)	Primary Key
UserType	int(10)	
Username	varchar(40)	

Password	varchar(40)	
EmailAddress	varchar(40)	
AdminComments	varchar(250)	

Dish

Column	Datatype	Notes
DishID	int(10)	Primary Key
Rating	int(1)	
Name	varchar(40)	
Description	varchar(250)	
Cuisine	varchar(40)	
Course	varchar(40)	
Diet	varchar(40)	
SpecialReq	varchar(100)	

AdminApprovesRestaurant

Column	Datatype	Notes
UserID	int(10)	Primary Key
RestaurantID	int(10)	Primary Key

UserMemberRegister

Column	Datatype	Notes
UserID	int(10)	Primary Key
UserID	int(10)	Primary Key

AdminApprovesMember

Column	Datatype	Notes
UserID	int(10)	Primary Key
UserID	int(10)	Primary Key

DishHasReview

Column	Datatype	Notes
DishID	int(10)	Primary Key
ReviewID	int(10)	Primary Key

DishIsPartOfRestaurant

Column	Datatype	Notes
RestaurantID	int(10)	Primary Key
DishID	int(10)	Primary Key

AdminEditsReview

Column	Datatype	Notes
UserID	int(10)	Primary Key
ReviewID	int(1)	Primary Key

UserReview

Column	Datatype	Notes
UserID	int(10)	Primary Key
ReviewID	int(10)	Primary Key

MemberReview

Column	Datatype	Notes
UserID	int(10)	Primary Key
ReviewID	int(1)	Primary Key

AdminReview

Column	Datatype	Notes
UserID	int(10)	Primary Key

ReviewID	int(1)	Primary Key

2.3 Design of Algorithm

To ensure that our system achieves its desired outcome of recommending dishes to users, we will need to create an algorithm that will recommend a dish to a user, based on a series of factors. These factors will be both their current preferences (ie. they are looking for a starter), and their previous preferences (ie. they dislike seafood, so any dishes containing seafood will not be recommended).

To allow for this to happen, our algorithm will look into the frequency at which specific dishes are chosen and also the ingredients contained in dishes that are often rated highly by the user. Using this information, we will be able to predict other dishes the user may like, and recommend these to the user.

The recommendation feature of our app adds a personalised touch so that each user feels like they are getting a unique experience. Different users will have different recommendations based on their own personal preferences, so users will feel like the app caters specifically to them.

In terms of the algorithm itself, the rough idea of it will entail:

If (like count of a dish is above the like threshold (as a percentage of total ratings))

User must like this dish

Look into the ingredients of this dish

Find other dishes the user has tried that contain same ingredients

See if user liked this dish

Infer rules for user likes/preferences

ie. if dish contains fish + rice -> user will like

Recommend dishes

See what other restaurants serve the same/similar dish

Recommend dishes

If (user dislikes a dish)

Find other dishes the user has tried that contain same ingredients

See if user liked this dish

Come up with rules for user likes/preferences

Ensure no similar dishes are recommended

2.4 Evaluation Design & Requirements of System

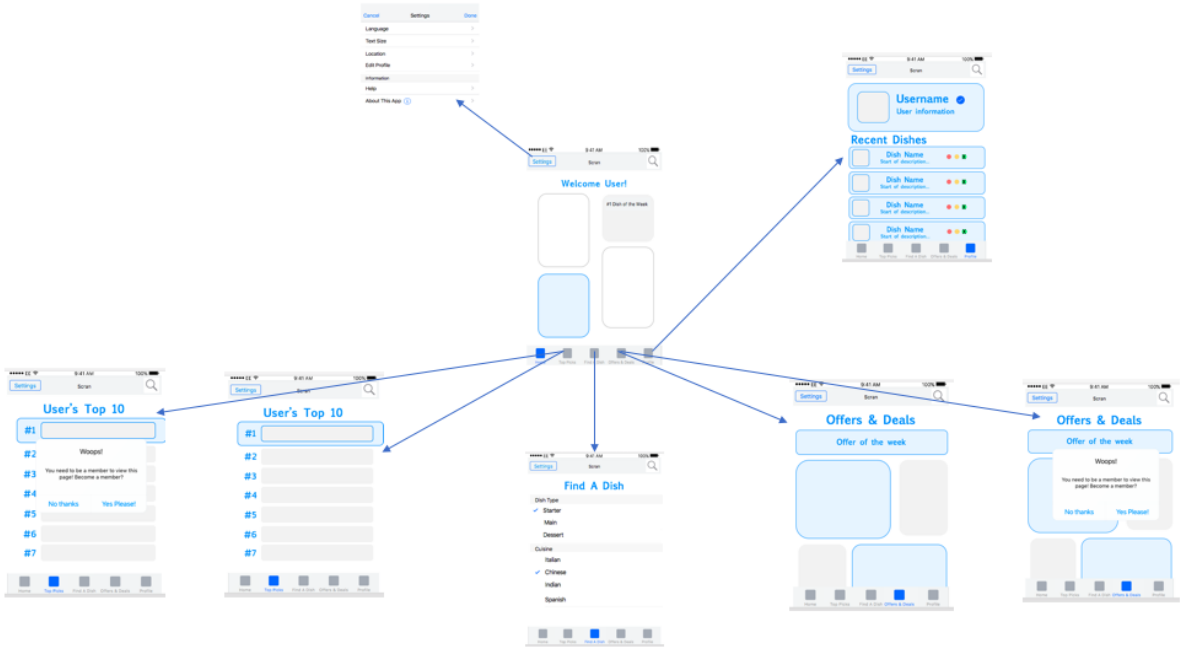
To evaluate our app and its system, we would initially test that all the essential features are working properly. This includes checking that :

- The UI map works perfectly, ie clicking on a button leads to the correct page corresponding to it.
- The speed time for loading images and other texts is acceptable.
- The search bar is able able to take strings as inputs and retrieve the relevant information from the database and displays the results properly.
- Validation of the search bar input, and preventing any possible SQL injection.
- The registered users receive a verification email to confirm their account creation with a valid URL.
- The app provides suggestions to the user based on his previously rated dishes. So we would create a sample profile and positively rate certains types of dishes to see if suggested dishes do correspond to the user's tastes and therefore see if the algorithms behind the app are fully functional.

2.5 User Interface Design

We have come up with some prototypes for the pages of our system. The actual colours used may vary slightly as we finalise what colour scheme we will go with, and some items on the page may move/be edited, however the general idea of what we want to include on each page is given. Below are some maps of how the pages would interact with each other, and there are bigger versions of each individual page given in the appendix.

The leftmost screen is the first screen the user will see when they open the app. From this screen they will have to either register for an account (see middle screen), or log in to an existing account (see rightmost screen).



The above map shows the main 'home screen' of the system and what pages can be reached from it. When logged in, the user will see the central screen and has various pages they can reach from it. There

will be a ‘settings’ button in the top left corner, which will give the user a choice of options to change about the app, such as the text size, language etc and they can also see information such as the ‘help’ section, or ‘about this app’ section.

There will be a menu of icons across the bottom of the pages (icons are represented with simple boxes at the current moment), which will link to the relevant page. The pages can all be accessed from within each other. The free user will be able to access icons 1, 3 and 5 (homepage, find me a dish and their profile) and will have locks over the icons for 2 and 4 (top picks and offers & deals). If they click on either of these pages, they will see a blank version of this page with a warning message on top of it. Members will be able to access all 5 pages.

For example, if a free user clicks on ‘top picks’ they will see the screen in the bottom left of the screen, whereas the members would see the screen next to it, which would contain their content. The content on such a page would include the user’s top choices and they would be ranked based upon how much they liked it, how many times they have chosen that dish etc.

Following the ‘top picks’ page, is the ‘find a dish’ page which would feature subsections such as ‘dish type’, ‘cuisine’ etc and the user can select what their preferences are at that moment. There would be more subsections than what is featured in the image, however for the purposes of the prototype we were only able to show 2 subsections so that the general idea of the page is given. This page features the main purpose of our system - allowing the user to give their preferences and recommending a dish for them, based on these preferences. Other subsections on this page will include “proximity to current location”, “restaurant/takeaway”, “spiciness” etc.

The next page is the ‘offers & deals’ page, which acts in a similar way to the ‘top picks’ page in the sense that members can access the page and view their personalised content, whereas free users will see a default/blank page with a warning message encouraging them to become members. This page will contain exclusive deals and offers only available to members, or priority alerts available for members. This page should offer an incentive for users to upgrade to become members as these offers will not be available to other users.

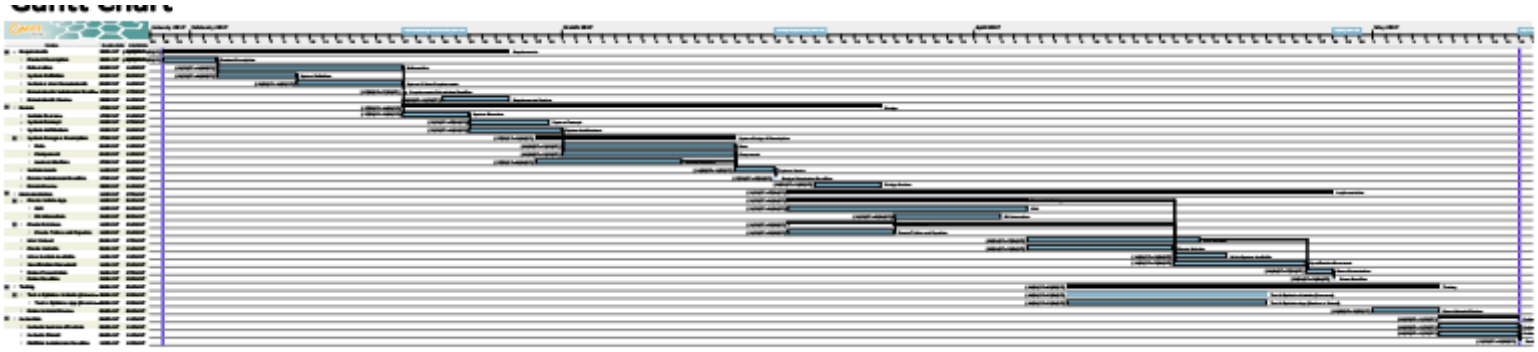
The final icon in the bottom menu is for the user’s profile, which will feature an image that they can upload (a default will be set if they do not choose one), the user’s username, a brief user information (ie their favourite restaurant, favourite dish etc), and if the user is a member they will have a tick icon next to their name. Below this section will be the user’s dish history, which will list the dish they had, the start of the description and what the user classified it as (loved it, liked it or hated it). We are unable to reach a conclusion at the moment as to the way in which the classification should be displayed at the moment, so have 3 different versions of displaying this. The various options for this page can be found in the appendix.

2.6 Logo Ideas

We have incorporated the logo into several different designs. We have yet to decide on which design we're going to choose however we have listed five of our favourite designs above. We tried to stick with warm colours when we were designing the logos as warmer colours are more inviting to users and they are more eye catching. We also decided to stick with a food-themed design and the chef hat was a popular choice with varying versions of it used.

3. Review Against Plan

So far, we have completed both the requirements and design stages of our project. This includes the majority of the planning for the project - who will complete each task, what subtasks our main tasks will be split up into etc. We have managed to complete the tasks so far in accordance with the Gantt chart, and have been able to meet both internal and external deadlines.
We do have some changes we have made to the original Gantt chart, as we had a new insight of what it needed to include once we had started the project. Our updated Gantt chart can be found below. Due to difficulty in making the chart visible on the page we have outlined the details of the Gantt chart in a table structure in the appendix.. We have also pasted the Gantt chart into separate sections to aid with visibility, which can also be found in the appendix.



4. Appendix

Use Case Descriptions

ID	UC1
Name	Access system
Actor Involved	Free User
Description	Accesses the "*Scran*" system either via the website or app
Pre-condition	System is running

ID	UC2
Name	Create free user account
Actor Involved	Free User
Description	The user creates an account that will allow them to access to free user privileges
Pre-condition	System is running
Inclusion points	UC3 - Create login details

ID	UC3
Name	Create login details
Actor Involved	Free User
Description	During the account application process the user must enter a username and password for their account
Pre-condition	System is running
Extension Points	UC4 - Account creation approved UC5 - Account creation not approved

ID	UC4
Name	Account creation approved
Actor Involved	Free User
Description	An administrator has approved the user's request to create an account and the user may now log in using this account
Pre-condition	System is running Administrator has approved request

ID	UC5
Name	Account creation not approved
Actor Involved	Free User
Description	An administrator has chosen to not approve this request
Pre-condition	System is running An administrator has checked the request and not approved it

ID	UC6
Name	Enter login details
Actor Involved	Free user
Description	The user enters the username and password belonging to their account
Pre-condition	System is running User has access to a valid account

ID	UC7
Name	Login rejected
Actor Involved	Free user
Description	The login details the user entered were incorrect and their login attempt has been rejected
Pre-condition	System is running User has access to a valid account

ID	UC8
Name	Apply for membership
Actor Involved	Free User
Description	The user makes a request to acquire a member account
Pre-condition	System is running
Inclusion points	UC9 - Create login details

ID	UC9
Name	Create login details
Actor Involved	Free User
Description	During the account application process the user must enter a username and password for their account
Pre-condition	System is running
Extension points	UC5 - Account creation approved UC6 - Account creation not approved

ID	UC10
Name	Account creation approved
Actor Involved	Free User
Description	An administrator has approved the user's request to create an account and the user may now log in using this account
Pre-condition	System is running Administrator has approved request

ID	UC11
Name	Account creation not approved
Actor Involved	Free User
Description	An administrator has chosen to not approve this request
Pre-condition	System is running An administrator has checked the request and not approved it

ID	UC12
Name	Browse restaurants recommendations by high ratings
Actor Involved	Free User
Description	Choose to view only a select number of restaurants that have a high enough rating
Pre-condition	System is running

ID	UC13
Name	Refine by cuisine/occasion
Actor Involved	Free User
Description	Choose to view only a select number of restaurants that match the rules based on cuisine/occasion specified by the user
Pre-condition	System is running

ID	UC14
Name	Search for restaurant
Actor Involved	Free User
Description	The user accesses the ‘Search Restaurants” function of the system
Pre-condition	System is running

ID	UC15
Name	Refine search by name/cuisine/GPS location
Actor Involved	Free User
Description	User refines their search by specifying a name/cuisine and/or allowing access to the GPS location of their device
Pre-condition	System is running

ID	UC16
Name	Select restaurant
Actor Involved	Free User
Description	The user selects a specific restaurant and is taken to the restaurant's page
Pre-condition	System is running

ID	UC17
Name	Rate restaurant
Actor Involved	Free User
Description	User rates the selected restaurant
Pre-condition	System is running

ID	UC18
Name	Browse dishes
Actor Involved	Free User
Description	The user browses the dishes of the selected restaurant
Pre-condition	System is running

ID	UC19
Name	Rate dishes
Actor Involved	Free User
Description	The user rates the dishes of the selected restaurant by either choosing that they “hated it”, “liked it”, or “loved it”
Pre-condition	System is running

ID	UC20
Name	Browse recommended dishes
Actor Involved	Free User
Description	The user browses the recommended dishes of the selected restaurant
Pre-condition	System is running

ID	UC21
Name	Select restaurant
Actor Involved	Member
Description	The user selects a specific restaurant and is taken to the restaurant's page
Pre-condition	System is running User has access to a valid account
Inclusion points	UC23 - Enter new dish

ID	UC22
Name	Enter new restaurant details
Actor Involved	Member
Description	The user enters the details of a new and unverified restaurant. This data entry attempt must then be approved by an administrator before it is properly added to the database
Pre-condition	System is running User has access to a valid account
Inclusion points	UC23 - Enter new dish

ID	UC23
Name	Enter new dish
Actor Involved	Member
Description	The user enters the details of a new dish from an unverified restaurant. This data entry attempt must then be approved by an administrator before it is properly added to the database
Pre-condition	System is running User has access to a valid account

ID	UC24
Name	Add comment
Actor Involved	Member
Description	The user adds a comment to the page of their selected restaurant. This comment must be approved by an administrator before it is made publicly viewable
Pre-condition	System is running User has access to a valid account

ID	UC25
Name	Edit unverified restaurant details
Actor Involved	Member
Description	The user edits the details of an unverified restaurant. This data entry attempt must then be approved by an administrator before it is properly added to the database
Pre-condition	System is running User has access to a valid account

ID	UC26
Name	Select dish
Actor Involved	Member
Description	The user selects a specific dish from the previously chosen restaurant and is taken to its dish page
Pre-condition	System is running User has access to a valid account

ID	UC27
Name	Add comment
Actor Involved	Member
Description	The user adds a comment to the page of their selected dish. This comment must be approved by an administrator before it is made publicly viewable
Pre-condition	System is running User has access to a valid account

ID	UC28
Name	Edit dish for unverified restaurant
Actor Involved	Member
Description	The user edits the details of a dish for an unverified restaurant. This data update attempt must then be approved by an administrator before it is properly added to the database
Pre-condition	System is running User has access to a valid account

ID	UC29
Name	Browse restaurants recommended by similar members
Actor Involved	Member
Description	The user selects to view only restaurants which have been given high ratings by other users which gave similar ratings to them
Pre-condition	System is running User has access to a valid account

ID	UC30
Name	Apply for restaurant account
Actor Involved	Restaurant
Description	The user makes a request to acquire a restaurant account (through a separate page to the ones used by free users and members
Pre-condition	System is running
Inclusion Points	UC31 - Create login details

ID	UC31
Name	Create login details
Actor Involved	Restaurant
Description	During the account application process the user must enter a username and password for their account
Pre-condition	System is running
Extension Points	UC32 - Account creation approved UC33 - Account creation not approved

ID	UC32
Name	Account creation approved
Actor Involved	Restaurant
Description	An administrator has approved the user's request to create an account and the user may now log in using this account
Pre-condition	System is running Administrator has approved request

ID	UC33
Name	Account creation not approved
Actor Involved	Restaurant
Description	An administrator has chosen to not approve this request
Pre-condition	System is running An administrator has checked the request and not approved it

ID	UC34
Name	Enter restaurant login details
Actor Involved	Restaurant
Description	The user enters the username and password belonging to their account
Pre-condition	System is running User has access to valid account Restaurant has been validated by administrator

ID	UC35
Name	Login rejected
Actor Involved	Restaurant
Description	The login details the user entered were incorrect and their login attempt has been rejected
Pre-condition	System is running

ID	UC36
Name	Add new dish
Actor Involved	Restaurant
Description	The user enters the details of a new dish from the restaurant they represent
Pre-condition	System is running User has access to valid account Restaurant is has been validated by administrator

ID	UC37
Name	Select dish
Actor Involved	Restaurant
Description	The user selects a dish from it’s own restaurant and is taken to its page
Pre-condition	System is running User has access to valid account Restaurant is has been validated by administrator

ID	UC38
Name	Edit dish
Actor Involved	Restaurant
Description	The user edits the selected dish from its own restaurant.
Pre-condition	System is running User has access to valid account Restaurant has been validated by administrator

ID	UC39
Name	Edit restaurant details
Actor Involved	Restaurant
Description	The user edits the name/location/opening times of the restaurant they represent
Pre-condition	System is running User has access to valid account Restaurant has been validated by administrator

ID	UC40
Name	Enter admin login details
Actor Involved	Administrator
Description	The user enters the username and password belonging to their administrator account
Pre-condition	System is running User has access to valid account

ID	UC41
Name	Login rejected
Actor Involved	Administrator
Description	The login details the user entered were incorrect and their login attempt has been rejected
Pre-condition	System is running

ID	UC42
Name	Approve restaurant/dish edits/additions
Actor Involved	Administrator
Description	The user grants permission to the member or restaurant representative to make the edits/additions they have tried to make
Pre-condition	System is running User has access to valid account

ID	UC43
Name	Approve flagged comments
Actor Involved	Administrator
Description	The user approves comments that have been flagged by other users
Pre-condition	System is running User has access to valid account

ID	UC44
Name	Approve restaurant account
Actor Involved	Administrator
Description	The user approves the application attempt made for a restaurant account
Pre-condition	System is running User has access to valid account

ID	UC45
Name	Review flagged member
Actor Involved	Administrator
Description	The user checks the account and account activity of a flagged free user/member/restaurant
Pre-condition	System is running User has access to valid account

ID	UC46

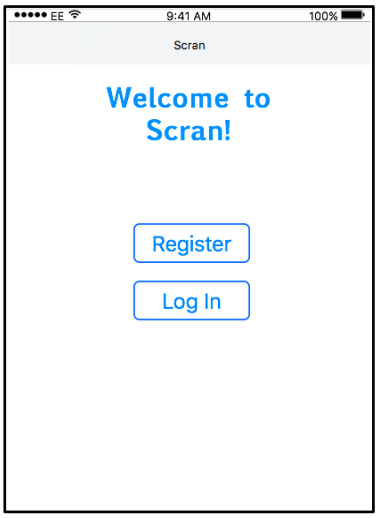
Name	Edit restaurant page
Actor Involved	Administrator
Description	The user edits the details contained on a restaurant page. This may include the name, location or opening times of the restaurant, and the dishes listed on the restaurant's page
Pre-condition	System is running User has access to valid account

ID	UC47
Name	Edit user rating
Actor Involved	Administrator
Description	The user edits the hidden rating belonging to a specific user which represents their activity on the system
Pre-condition	System is running User has access to valid account

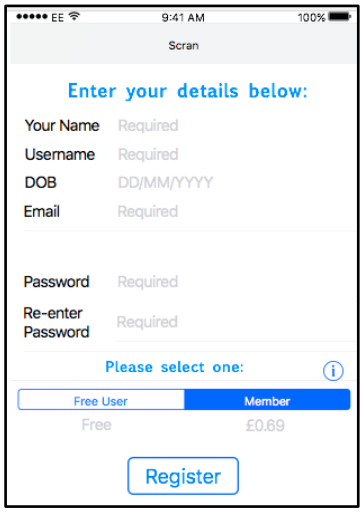
ID	UC48
Name	Block/ban members/restaurants
Actor Involved	Administrator
Description	The user blocks/bans a member or restaurant account, likely due to repeated rule breaking
Pre-condition	System is running User has access to valid account

User Interface Pages

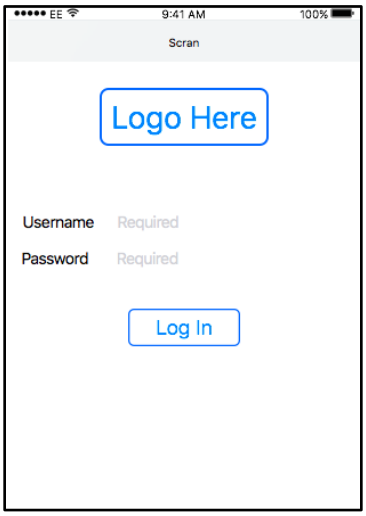
Initial Page



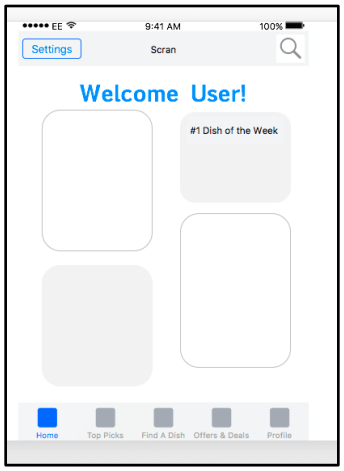
Register



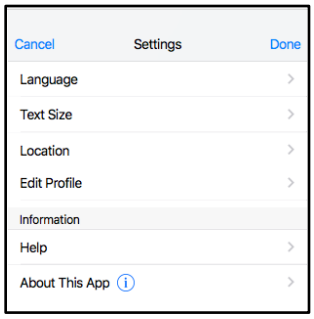
Log In



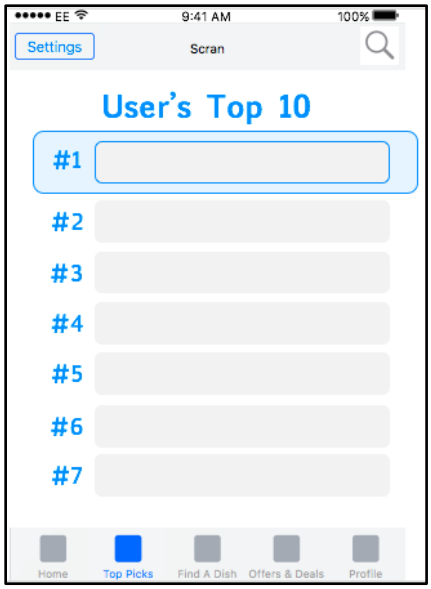
Home Page



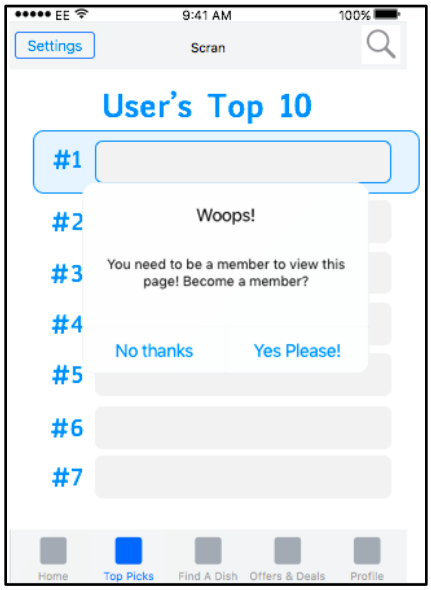
Settings Menu



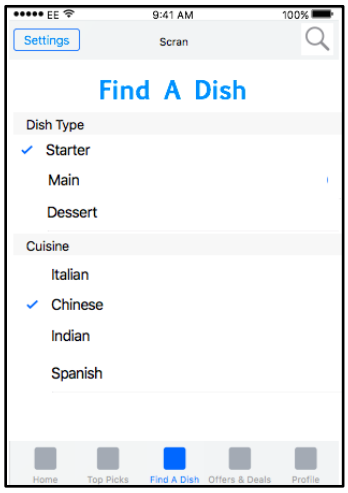
Top Picks Page (Member View)



Top Picks Page (Free User View)



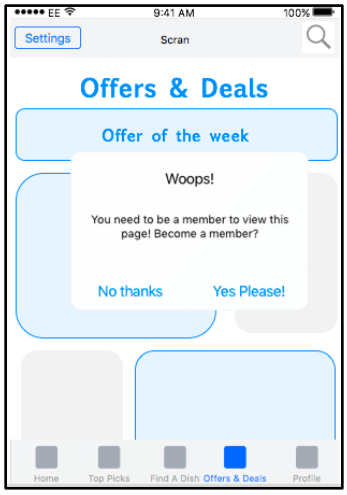
Find A Dish Page



Offers & Deals

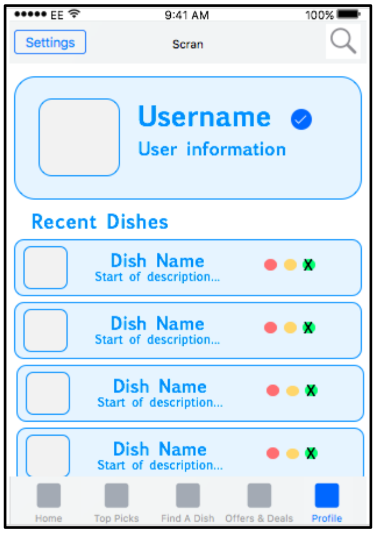
(Member View)

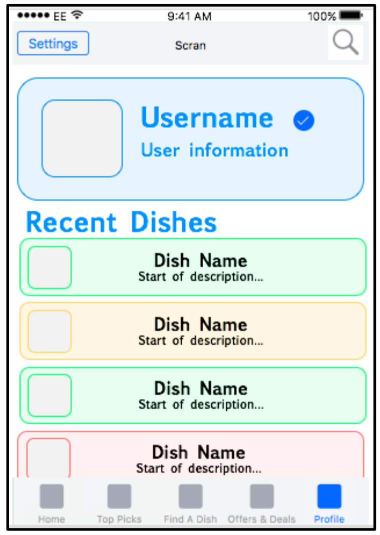
Offers & Deals

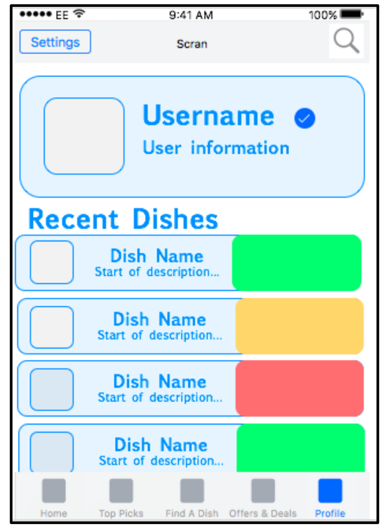


(Free User View)

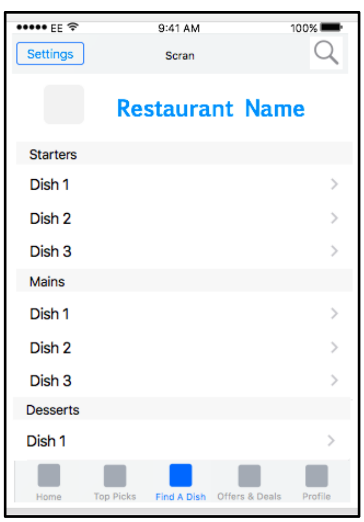
Ideas for Profile Page



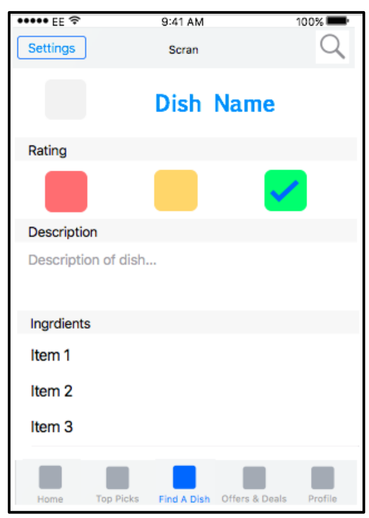




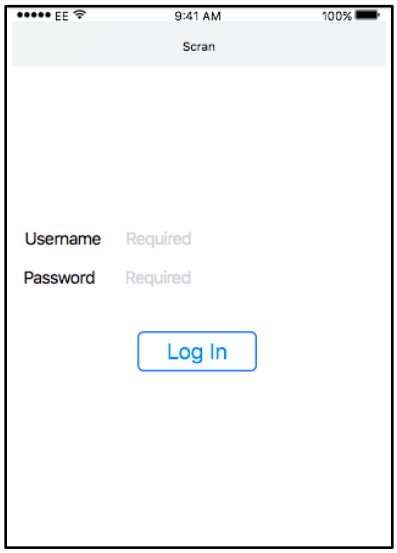
Restaurant Menu Page



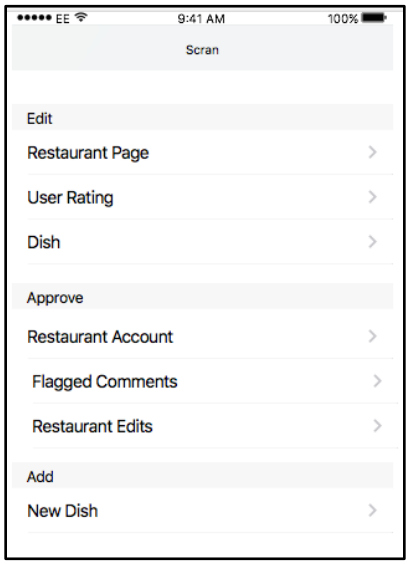
Dish Page



Admin Login Portal



Admin Home Screen



Data Dictionary

Word	Definition
Free User	A free user is a user who uses the 'free' version of the app, where they have to make an account and their preferences can be tracked, but they don't have access to certain extended features. To access these features, they need to become a "Member" (see below), and any free user can become a 'Member' if they choose to.
Member	A member is an upgraded version of a 'free user' (see above), and uses a version of the app that has all features unlocked. They still need to make an account so that their preferences can be tracked, but if they have already been a 'free user' and upgrade to be a 'member', the same account can be used, they will just be changed to a 'member'.
Restaurant (User)	A restaurant (user) is a user who represents a restaurant that is advertised on the app. There still needs to be an account made for this user so that their information can be stored, and they can be sent feedback regarding their dishes, but they would not use the app in the way that a normal user would. They would not search for restaurants and dishes, and leave ratings on them, but if we have time to include the comments feature, a restaurant would be able to reply to comments left from other users.
Administrator	A user who is in charge of maintaining the system. They will be allocated full access to all database data including the ability to edit and remove data.
UserID (FreeUser, Member & Administrator)	This is the primary key of the FreeUser table, Member table and Administrator table. Values are unique across these three tables to allow a free user to become a member, and they would retain the same UserID.
UserType (FreeUser, Member & Administrator)	This is either 'free user', 'member' or 'administrator'.
Username (FreeUser, Member & Administrator)	This is the name chosen by the user when they register for the app, and will be how they are identified. It will be featured on their profile, it will be used when they login to the app and should also be unique.
Password (FreeUser, Member &	This is chosen by the user when they register for the app and will be used along with their username (above) when they login. It will be used for authentication of

Administrator)	user, and should be min. 8 characters long with a combination of an uppercase letter, lowercase letter and number.
EmailAddress (FreeUser, Member & Administrator)	This is the email address provided by the user when they register for the app. Any communication required with the user will be sent to this address, including adverts, promotions etc (if the user consents to this).
AdminComments (FreeUser, Member & Administrator)	This is used to give a brief explanation by Administrators of a user's actions or reasons for any UserRating edits or request denials.
RestaurantID (Restaurant)	This is the primary key of the Restaurant table and will be a unique value for each restaurant.
RestaurantName (Restaurant)	This is the name of the restaurant that will be used to identify where the dish is served.
Latitude (Restaurant)	This is the value of the latitude of the position of the restaurant, and will be used when recommending restaurants within close proximity to the user.
Longitude (Restaurant)	This is the value of the longitude of the position of the restaurant, and will be used when recommending restaurants within close proximity to the user.
RestaurantRating (Restaurant)	This is the value of the restaurant as rated by users.
Verified (Restaurant)	A restaurant not added by users - added by an owner or representative of the restaurant.
DishID (Dish)	This is the primary key of the Dish table and will be a unique value for each dish featured on the app.
Rating (Dish)	Average rating of the dish at that particular restaurant by users.
Name (Dish)	This is the name of the dish and will be used in the description of the dish on the app. The user can search for a dish by its name among other attributes.

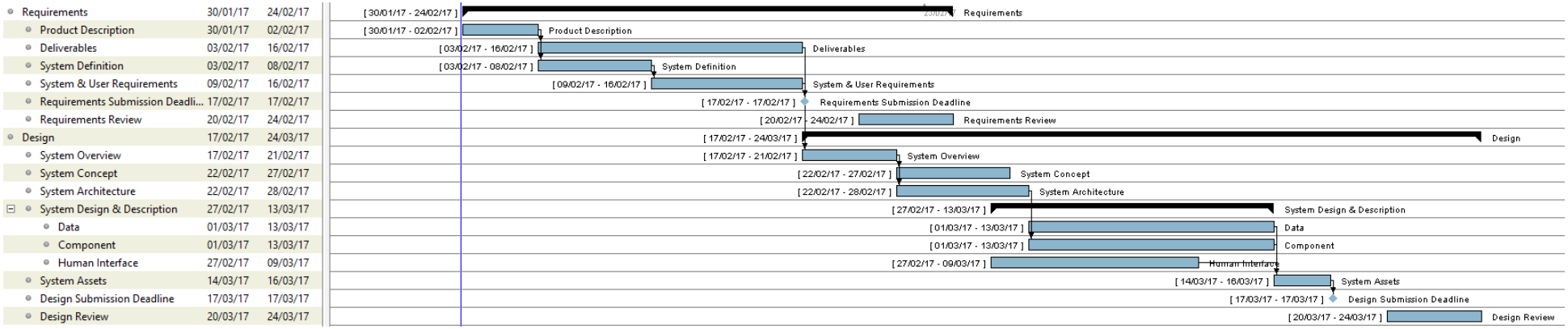
ReviewID (DishReview)	This is the primary key of the DishReview table and will be used to uniquely identify each review left.
UserID (DishReview)	This is a foreign key in the DishReview table, and refers to the user who left the review.
DishID (DishReview)	This is a foreign key in the DishReview table, and refers to the dish the review is about.
Comment (DishReview)	This is the actual review left about the dish (ie the comment left by the user).
Flagged (DishReview)	Boolean - True or False (default false) If flagged by another user, then set to true to be reviewed by admin.
Karma (FreeUser and Member)	Value hidden from user. Used by admins to determine whether a user is behaving poorly - e.g. having a comment rejected -1, if comment reviewed by admin to be prejudice, then -10.
Portal	Synonymous with viewpoint Admin portal - Admin website User portal - User website/ app Restaurant portal - Webpage on user website/ app

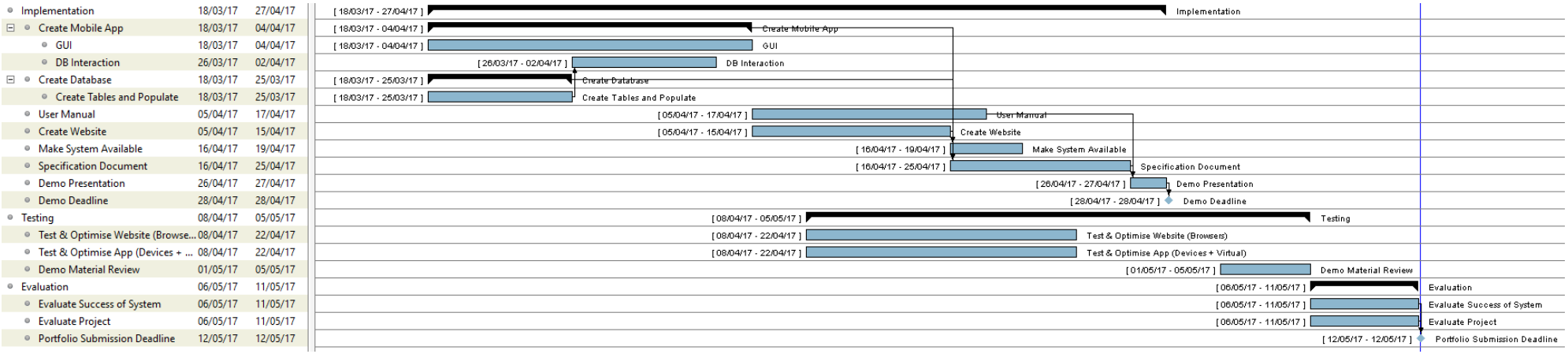
Gantt Chart Details

Task / Sub-task	Start Date	End Date	Completed By
Requirements	30/01/17	24/02/17	All
Product Description	30/01/17	02/02/17	Danny
Deliverables	03/02/17	16/02/17	Kris
System Definition	03/02/17	08/02/17	Eklavya
System & User Requirements	09/02/17	16/02/17	Lucy & Tom
Requirements Submission Deadline	17/02/17	17/02/17	All
Requirements Review	20/02/17	24/02/17	All
Design	17/02/17	24/03/17	All
System Overview	17/02/17	21/02/17	Lucy & Kris
System Concept	22/02/17	27/02/17	Tom
System Architecture	22/02/17	28/02/17	Danny & Eklavya
System Design & Description	27/02/17	13/03/17	Lucy, Kris & Eklavya
Data	01/03/17	13/03/17	Danny
Component	01/03/17	13/03/17	Tom
Human Interface	27/02/17	09/03/17	Lucy
System Assets	14/03/17	16/03/17	Kris & Eklavya
Design Submission Deadline	17/03/17	17/03/17	All

Design Review	20/03/17	24/03/17	All
Implementation	18/03/17	27/04/17	All
Create Mobile App	18/03/17	04/04/17	All
GUI	18/03/17	04/04/17	Lucy & Tom
DB Interaction	26/03/17	02/04/17	Kris, Danny & Eklavya
Create Database	18/03/17	25/03/17	Danny & Eklavya
Create Tables and Populate	18/03/17	25/03/17	Kris
User Manual	05/04/17	17/04/17	Lucy & Tom
Create Website	05/04/17	15/04/17	Kris, Danny & Eklavya
Make System Available	16/04/17	19/04/17	Lucy & Tom
Specification Document	16/04/17	25/04/17	All
Demo Presentation	26/04/17	27/04/17	All
Demo Deadline	28/04/17	28/04/17	All
Testing	08/04/17	05/05/17	All
Test & Optimise Website (Browsers)	08/04/17	22/04/17	Kris, Danny & Eklavya
Test & Optimise App (Devices + Virtual)	08/04/17	22/04/17	Lucy & Tom
Demo Material Review	01/05/17	05/05/17	All
Evaluation	06/05/17	11/05/17	All
Evaluate Success of System	06/05/17	11/05/17	All
Evaluate Project	06/05/17	11/05/17	All
Portfolio Submission Deadline	12/05/17	12/05/17	All

Expanded Gantt Chart





Use Case Diagram