CODEV

FUNCTIONAL SPECIFICATION

MEng Year 3

Department of Electronics

University of York

Software Engineering Group 4

Document Control

Version	Modified By	Date	Section Modified	Remarks
1	James P, James C, & Nathan B	17/01/20	All	
1.1	Louis Newman	19/01/20	Risks	
1.2	Joe Butler	20/01/20	Risks	
1.3	James P & Becky	30/01/20	All	

Table of Contents

1. Product Overview

The Scran Plan cooking application aims to help users batch cook meals on a budget. The user can select and collate recipes that they plan to cook for that week and create an in app shopping list that is constructed from the ingredients that the user specified they do not own. With the added ability of creating a personalised profile, the user will be able to tailor the application to their tastes so that they can filter out any of the recipes that don't suit, save their favourite recipes and share them with their friends.

2. Functional Specification

2.1 Essential features

Feature	Description	Rationale
Launch Screen (1.1) User Stories: A1, A2 , C20	Available for first time users. Guides users through account creation and customisation, setting them up for streamlined use of the application	Easy process for setting first time users up for a comfortable experience with the application
Recipe home (1.2) User Stories: C1, C2, C3, C6, C7, C18	Once users have logged in, takes them to the main home screen with all the main recipes. From here they can access various menus and their profile	Quickly taken to main recipe page for the user to view recipes. Easy access from there to other parts of the application
Accounts (1.3) User Stories: A3, A4, A5, A11	Users can create, edit, login and delete accounts from which they will rate, save and create recipes on the application	Gives users their own custom space on the application from which they can interact with features and other users of the application
Privacy options (1.3) User Stories: A3, A4, A5, A11	Users can customise the information that is displayed on their account when viewed by other users	Essential for user privacy and allows users to feel safe and secure while using the application
Recipe presentations (1.4)	Pressing a 'Lets Cook' button next to each recipe will launch a presentation detailing the steps the user will follow to cook the meal. Each slide can contain multimedia and text elements to guide	Simplifies the cooking process of the meal and allows the user to concentrate on a single step at a time while following

User Stories: B1, B2, B3, B4, B6	the user through the task	the recipe
Meal planner (1.5) User Stories: C4, C5, C14, C17, C21, C22, C34	A weekly diary containing meals that the user has saved and assigned to a time slot for each day. Every recipe will have the ‘Let's Cook’ button next to it.	Allows the target audience of students to plan their meals effectively and easily launch the recipe from the planner directly
Recipes (2.1) User Stories: A12	Users can create, edit and delete recipes for meals they want to share. Each recipe will contain information such as pricing and difficulty	The primary function of the application, gives users the content that they primarily use the application for
Recipe ratings and reviews (2.2) User Stories: A8, C9, C11, C23, C30	Every recipe can be rated and reviewed by users using a 5 Star rating system and a more in-depth review system which would allow written reviews with pictures and a recommendation rating. Other users can vote on the helpfulness of reviews so that more relevant reviews are shown to users	Allows users to vote on recipes that they want to see on the application are removes spam/inadequate recipes from showing on the application
Search bar (2.3) User Stories: C13	Users can search for recipes with keywords such as ingredients or method	Makes finding particular recipes easier
Recipe filters (2.3) User Stories: C13	Users can filter recipes to only show particular things they are looking for, such as excluding/including particular ingredients, conforming to a religious diet or being vegan or vegetarian. Additionally, this can be extended to an account wide filter, only allowing recipes of a chosen category to ever be shown. User will also be able to sort recipes through several categories, such as price or ratings	Certain users will require certain dietary/religious needs and will not be interested in recipes that they will not be able to follow. This will allow users to find recipes more suitable for them based on price or quality of meal
Saving recipes (2.4) User stories: C8, C15	Users can save recipes they like and access saved recipes through a button on the application	Allows easy access to recipes users would like to follow at a later date
Healthy living system	Tracks users food intake and displays nutritional information about their diet based on recipes	Keeps users informed of their eating habits

(2.5) User Stories: A13, C31, C32	they have followed on a weekly basis	
Recipe and ingredient information (2.6) User Stories: C12	Metadata is attached and displayed for each ingredient and recipe, showing information such as nutritional value, price, typical expiration date etc.	Gives users an in-depth summary of particular ingredients and recipes they are interested in
Gold membership (Get rid of ads) (2.8) User Stories: A9, C19	Allows users to upgrade to gold membership to remove adverts from within the application	Removes adverts for the user while still bringing in income

2.2 Configuration

2.2.1 Account Configuration

Event	Configurable Elements	Rational behind configurable elements
Account creation by user so that a personal profile can be built	Email* Password* Date of Birth*	Users should be able to add their account details initially when entering the app. They should be able to change this at a later date within account settings.
Account settings can be edited by the user at any time in order to change the users preferences to suit the app	Email* Password* Date of Birth* Dietary requirements/preferences Privacy Settings Notifications Upgrade of account Account Deletion	Users should be able to change their account settings after their initial setup. They should be able to change their privacy settings for each member can decide how much other members can see. Each member should be able to add their dietary requirements which will help to filter recipes later in the program. They should also be able to toggle notifications and if they would like to upgrade/downgrade their account.
Account settings notifications controlled by user	Notifications on/off	Users should be able to turn notifications on or off at any point in the product lifespan.

Account settings and privacy settings adjusted by user	Public Private Custom	Users should be able to change their privacy settings at any time. From public where all information apart from: Email and passwords are displayed. To private where nothing is displayed about this user to other people. The user should also be able to customise their settings so they can be somewhere in-between these extremes.
Account settings and dietary requirements set up by the user	List of common allergies/ food intolerances	Users should be able to add their dietary requirements/ any food intolerances they have to allow recipes to be filtered.
Account settings, the ability to upgrade the account by the user	Upgrade to premium features	Users should be allowed to upgrade to premium features of the app for a price.

2.2.2 Recipe Configuration

Event	Configurable Elements	Rational behind configurable elements
Adding user recipe	Ingredients Recipe instructions	All content for the recipes should be user configurable including the steps and the ingredients needed in the recipe.
Deleting user recipes	All recipe details	Users should be able to delete their own recipes together with on account deactivation recipes by the users should be removed.
Formatting recipes after initial creation by user	All recipe details	Users should be able to format their recipes after creation and change details to the methodology.

2.3 Non-essential Features

Feature	Description	Rationale
Different measurement systems for ingredients (2.7) User Stories: A14, C12	Users can customise the units of measurements that ingredients are displayed in, such as imperial, metric or cups	Allows users to comfortably display amounts of ingredients they need in a format they understand
Notifications (3.1) User Stories: A6, A7	The application can send push notifications to the user to alert them of a variety of things, such as upcoming expiry dates of ingredients/meals and reminders of their planned meals through the meal planner	Reminds users of several things without them having to open the application
Chef Stars and Chef Badges (3.2) User Stories: A8, A10	Users are awarded Chef Stars based on the overall 5 Star rating system for recipes they have submitted. Badges are also awarded for achievements earnt through use of the application	Gives users a sense of accomplishment and pride of their use of the application, allowing them to display their achievements on their profile
Icons for ingredients and recipes (3.3) User Stories: A14,	Each ingredient and recipe will display icons displaying certain information, such as whether the ingredient is vegetarian, vegan, halal etc.	Quickly displays important information to the user at a glance
Reporting of content (3.4) User Stories: C23	Offensive and inappropriate content can be reported by a user and removed from the application	Allows users to remove content they deem offensive/inappropriate
Recipe suggestions (3.5) User Stories: B5, C25	Suggests recipes to users based on ingredients they frequently use or the categories of recipes they have recently viewed	Allows users to discover new recipes that they will like and not necessarily have discovered
Social media (3.6)	Allows users to add/remove friends and share recipes they recommend onto their timeline for	Encourages users to use the application more and

User Stories: C16, C26, C27, C28, C29, C31	other users to view. Users will be able to see their friends activity on the application	recommend it to their friends

3. Risks

Risk factors that could affect the functional design of the project are as follows:

Risk **(Likelihood / Impact measured out of 10)**	**Description**	**Counter**
Server data structure efficiency likelihood: 3 Impact: 9	All downloads from the server will need to be filtered to the customers demands. Database and XML server data will need to be organized and include all relevant fields for the data to meet all demands for compound querying. Not getting this done correctly the first time could potentially have effects on code efficiency, meaning that making changes later will slow down the development process bringing things to a halt while potentially having to rewrite any code for other modules. On top of stopping the development process, working hours will need to go up to correct the issue meaning more costs for the company. Another reason to keep all queries clear, careful and concise is that when we move from the free storage to the pay as you go storage in the Firestore; we do not want to enter any infinite loops or over charge ourselves with over querying.	Input serious consideration when designing the data structure and fields to go with each document to take advantage of all querying potential (inequalities, equalities, array-contain, array-contains-any and in), to give the customer the most relevant information first while still limiting the information downloaded from the server to an ‘only when we need more’ basis. As all data will be ordered through inequality based querying there will be no need to sort on the client side.

Reaching server document limits likelihood: 2 Impact: 10	Each document within the firestore has a maximum size of 1Mb and fewer than 100 fields. Usually this would not be an issue but we will be using duplication to duplicate recent activity of users that the current user is friends with to create a social feed for each user.	A limit will be put on how manys posts can be present in any one users social feed. Along with this we can limit the amount of data taken from each event by only showing a preview of what the activity post contains. If this becomes a problem we may need to scale this up to multiple documents as the app grows.
Reaching server host limits likelihood: 4 Impact: 10	To save on costs a free server provider (Firebase), will be used with a maximum server network throughput of 10gb/month and a total storage capacity of 1gb. Both of these limits means on top of the file limit efficient XML server code will be required and network traffic will need to be restricted or monitored if the network usage is getting too close to the limit per user and during development testing. If these limits are reached the risk is that we will need to spend additional funds to pay for additional network storage and capacity.	Reduce XML file size wherever possible by finding the most efficient way to store/retrieve documents in a way that equally balances server storage and bandwidth efficiency and adjust accordingly depending on which is more important as development progresses. Create short but readable XML markup names and limit the total number of characters that can be inputted for a string query. Ensures that we can reduce our overall file size and prevent users from spamming long character strings to overload our server capacity. Make estimates for the total number of users the platform can support based off current usage figures of our own testing team. If the total number of users that can be supported turns out to be too low, evaluate if expanding to a paid server plan is economically favourable or not. May introduce measures to prevent users from sending to many requests simultaneously and limit network download/upload speed from interaction with UI app elements that require a lot of server data to be retrieved. This will also prevent bots and can be a

		good method of differentiating paid or free plans with limits on overall network traffic being enforced.
Feature creep happening likelihood: 8 Impact: 3	Towards the end of each iteration there is likely to be a desire to push for further functionality before launching each iteration to improve the flow of the app.	Ensure only stories that are tasked are completed and any need to add in elements of other stories in order to complete a current story is explicitly specified so any changes to the iterations are made to ensure deadlines are met.
Hardware implementation and testing on different devices likelihood: 4 Impact: 10	The android platform has a very wide ecosystem of smartphones & tablets it can run on. If proper tests for different devices aren't simulated and if in production we assume falsely that every device, new or old will be able to be supported. Issues will occur between different hardware implementations of the same code and the app may not be able to run at all on certain devices at launch or be limited in its overall performance.	All team members will use Android Studio v3 to develop. Android Studio tries to guarantee Hardware support down to a minimum version of Android on all devices that can be simulated within the application. Alongside this tests will still need to be done on the maximum amount of actual physical Android devices possible to ensure that functionality isn't limited or restricted between different manufacturers and models, especially in evaluating the performance and speed of the UI and wireless connectivity of the device in conjunction with the app. For this reason each team member who owns an android device with an Android version above the minimum project version will be asked to use there own device to test the code in addition or instead of the Android SDKs simulators.
Versioning likelihood: 2 Impact: 10	Differences in development versions/platforms or differences in the source code versions or branch of the code stored locally on a team members computer, can be significantly different and cause	To ensure all team members are developing using the same resources for the same version, only Android Studio V3 will be used to develop the app on a minimum supported Android

	issues when combined into a single working version.	Version of KitKat 4.4 (95.3% support). The only exception to this is the XML server code. This is standardized and provided the same version of XML is used can be tested in any modern text editor with XML support Github versioning is also used to make sure a version of the source code stored locally on a team members computer is the same as everyone else's. Any additions from another member are updated upon a pull request being received after code has been reviewed. Github branches also allow for development to occur parallel to the main release code for the purpose of safe testing.

4. Iterations

4.1 First Iteration

The First iteration of the application will focus on having a stable, working version with the main core features:

- (1.1) Launch screen with user login or profile creation options
  - User Stories: A1, A2 , C20
- (1.2) Main home screen where the recipes will displayed
  - User Stories: C1, C2, C3, C6, C7, C18
- (1.3) User profiles with privacy options enabled
  - User Stories: A3, A4, A5, A11
- (1.4) Launch a presentation detailing the steps the user will follow to cook the meal
  - User Stories: B1, B2, B3, B4, B6
- (1.5) Enabled meal planner
  - User Stories: C4, C5, C14, C17, C21, C22, C34

4.2. Second Iteration

The second iteration will include the important features that are not apart of the core foundations of the application. These features are:

- (2.1) User created recipe
  - User Stories: A12
- (2.2) Recipe ratings and reviews
  - User Stories: A8, C9, C11, C23, C30
- (2.3) Recipe sorting that includes filters and search functionality
  - User Stories: C13
- (2.4) Save favorite recipe option
  - User stories: C8, C15
- (2.5) Healthy living system to track calories and nutritional information
  - User Stories: A13, C31, C32
- (2.6) Recipe and ingredient information
  - User Stories: C12
- (2.7) Measuring systems for the recipes
  - User Stories: A14
- (2.8) Gold membership (Get rid of ads)
  - User Stories: A9, C19

4.3 Third iteration

The third iteration will include more of the extended functionality of the app:

- (3.1) Notification system
  - User Stories: A6, A7
- (3.2) Chef starts and Chef badges
  - User Stories: A8, A10
- (3.3) Icons for ingredients and recipes
  - User Stories: A14
- (3.4) Reporting of content
  - User Stories: C23
- (3.5) Recipe suggestions
  - User Stories: B5, C25
- (3.6) User timeline that allows user to display and share cooked meal recipes to friends.
  - User Stories: C16, C26, C27, C28, C29, C31

Functional Specifications Document

Authorization Memorandum

I have carefully assessed the Functional Specifications Document for the Scran Plan Application. Based on our authority and judgment, the continued operation of this system is authorized.

____________________________________ ________________________

NAME DATE

Project Leader

____________________________________ ________________________

NAME DATE

Customer