RECIPE FOR MEALS

PROJECT PLAN

İbrahim Türkmen 150140002
Ali Osman Atik 150140804
Emre Yeniay 150110013
Baran Kaya 150130032
Caner Işık 150130023

OCTOBER 10, 2017

İTÜ

Table of Contents

1. Introduction

Recipe4Meals is an Android-platform mobile application that provides users with lists of easy-to-understand recipes and suggests several suitable dishes that the user can cook given the user`s inventory. The application asks for the user`s inventory and then it will be saved into their profile, every time the user selects a meal to cook the program reconfigures the inventory list. The main goal of this project is to help people cook exotic or different meals for themselves, giving them a huge sense of accomplishment. The other aim is for bachelors to survive on their own without becoming dependent on their mothers.

1.1. Project Scope

With the help of Recipe4Meals, users can:

- Keep track of their inventories.
- Cook exotic dishes from different or the same cultures
- Rate and review their foods
- Can decide on what their next meals would be
- List dishes in alphabetical, difficulty or score order
- Filter dishes by their cultural, meal-they-belong-to, time it takes to cook or the user`s favorites
- Can see which dishes they can cook with their inventory
- Search for foods by their score, ingredients, name and cultural

1.2. Deliverables

Number	Deliverable	Due Date
1	Project Plan	13/10/2017
2	Requirements Document	25/10/2017
3	Design Document	30/11/2017
3.1	Database Design	10/11/2017
3.2	Searching and Filtering	20/11/2017
3.3	Recipe Page Design	30/11/2017
4	Application Code	12/12/2017
4.1	User API	12/12/2017

1.3. Epics

Number	Epic	Matched Deliverable	Size
1	Database Connection	3.1 Database Design	XL
1.1	Recipe Data	-	M
1.2	Review Data	-	S
1.3	Rating Data	-	S
2	Searching	3.2 Searching and Filtering	L
2.1	By name	-	T
2.2	By rating	-	T
2.3	Sorting	-	M
2.4	Filtering by cuisine	-	T
2.5	Filtering by meal time	-	T
2.6	Filtering by inventory	-	T
3	Recipe Pages	3.3. Recipe Page Design	M
3.1	Step-by-Step page	-	S
3.2	Cooking time	-	T
3.3	Difficulty	-	T
3.4	Cuisine	-	T
3.5	Dish Pictures	-	T
4	Interaction with user	4.1 User API	XL
4.1	Leaving Comment	-	L
4.2	Rating	-	M

1.4 Non-functional Issues

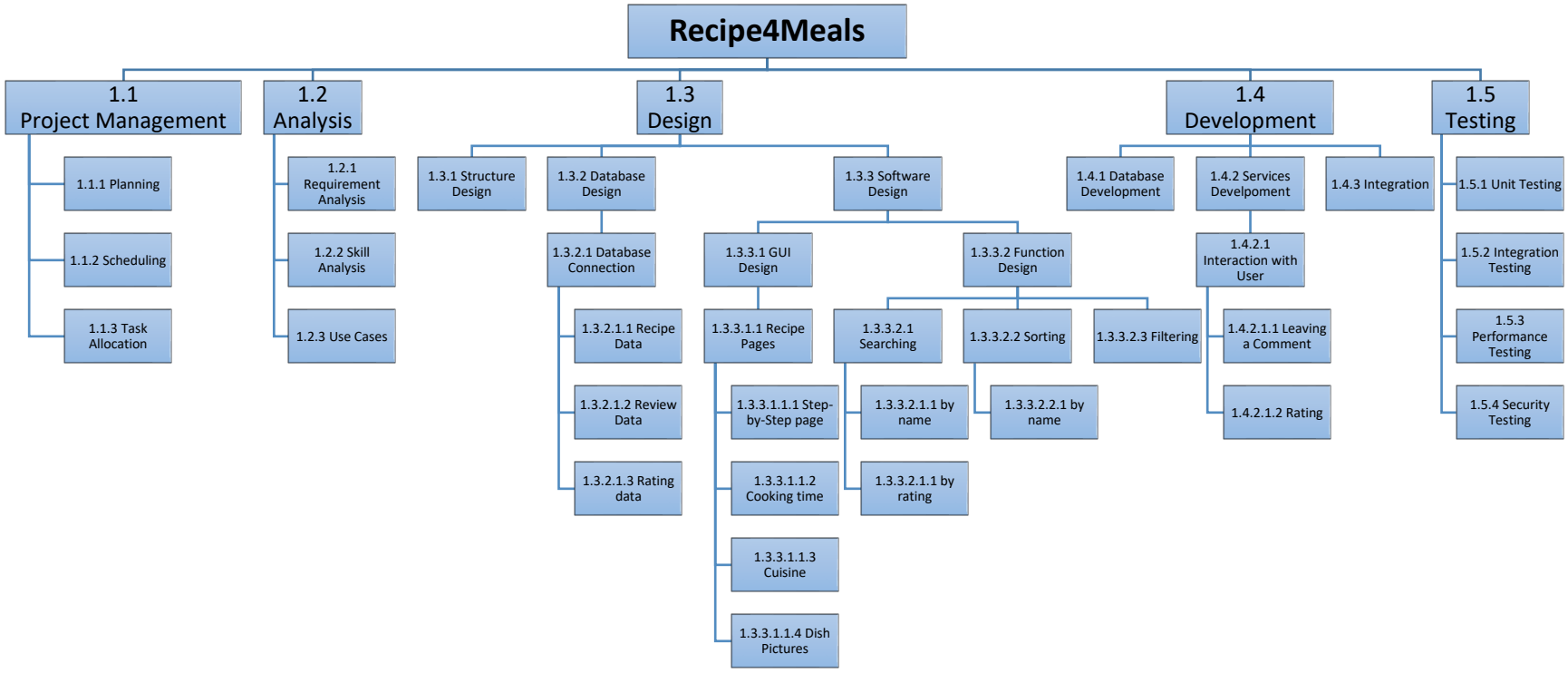
- **Performance:** Since this application, does not have any complicated computations, performance problems related to calculations is not an issue.

- **Portability:** Recipe4Meals application is planned as an Android application and it is an issue that it will not be present in iOS or web browsers.

- **Usability:** The application will be very simple and front-end team will develop the application in a user-friendly way.

- **Security:** This application will not collect any important or personal user information. Therefore, there is not an important security issue.

2. WBS


Recipe4Meals
1.1 Project Management
1.1.1 Planning
1.1.2 Scheduling
1.1.3 Task Allocation
1.2 Analysis
1.2.1 Requirement Analysis
1.2.2 Skill Analysis
1.2.3 Use Cases
1.3 Design
1.3.1 Structure Design
1.3.2 Database Design
1.3.2.1 Database Connection
1.3.2.1.1 Recipe Data
1.3.2.1.2 Review Data
1.3.2.1.3 Rating data
1.3.3 Software Design
1.3.3.1 GUI Design
1.3.3.1.1 Recipe Pages
1.3.3.1.1.1 Step-by-Step page
1.3.3.1.1.2 Cooking time
1.3.3.1.1.3 Cuisine
1.3.3.1.1.4 Dish Pictures
1.3.3.2 Function Design
1.3.3.2.1 Searching
1.3.3.2.1.1 by name
1.3.3.2.1.1 by rating
1.3.3.2.2 Sorting
1.3.3.2.2.1 by name
1.3.3.2.3 Filtering
1.4 Development
1.4.1 Database Development
1.4.2 Services Develpoment
1.4.2.1 Interaction with User
1.4.2.1.1 Leaving a Comment
1.4.2.1.2 Rating
1.4.3 Integration
1.5 Testing
1.5.1 Unit Testing
1.5.2 Integration Testing
1.5.3 Performance Testing
1.5.4 Security Testing

3. Estimates

1. (ILF) Internal Logical Files :
    - (10) Recepies table
    - (7) Ingridients table
    - (7) Cuisine table
    - (7) Comments & Ratings table
    - (10) Images table
    - (7) Cookies file
2. (EIF) External Interface Files :
    - (5) Adding comment
    - (5) Giving a rating
3. (EI) External Inputs :
    - (3) Select by type screen
    - (3) Select by cuisine screen
    - (3) Select by ingridients screen
    - (4) Comment & rate screen
    - (4) User interaction buttons
4. (EO) External Outputs :
    - (4) Main screen
    - (4) Selected type screen
    - (4) Selected cuisine screen
    - (4) Selected type screen
    - (4) Matching recepies for ingridients screen
    - (5) Comment & rate send screen
    - (5) The recepie screen
5. (EQ) External Queries :
    - (3) Recipies list
    - (3) Ingridients list
    - (3) Cuisine list
    - (3) Type list
    - (4) Search by type/cuisine/ingridients list
    - (4) Filter/sort results list

Type of Component	Complexity of Components			
	Low	Average	High	Total
EI	3 *3	2 *4	0 *6	17
EO	5 *4	2 *5	0 *7	30
EQ	4 *3	2 *4	0 *6	20
ILF	4 *7	2 *10	0 *15	48
EIF	2 *5	0 *7	0 *10	10
Unadjusted Function Points				125

GSC	Values (0-5)
Data communications	4
Distributed data processing	3
Performance	3
Heavily used configuration	2
Transaction rate	2
On-Line data entry	2
End-user efficiency	5
On-Line update	4
Complex processing	2
Reusability	5
Installation ease	5
Operational ease	4
Multiple sites	1
Facilitate change	2
	Total = 44

VAF = 0.65 + (Sum of all GSC factors) * 0.01

VAF = 0.65 + 44*0.01 => VAF = 1.09

Adjusted FP = VAF * (Total Unadjusted FP)

Adjusted FP = 1.09 * 125 => Adjusted FP = 136.25

We are assuming the programs that we need to use for this project has an average rate of 50 LOC/ FP and each programmer works for 20 FP per week. Then,

136.25 * 50 = 6812 LOC

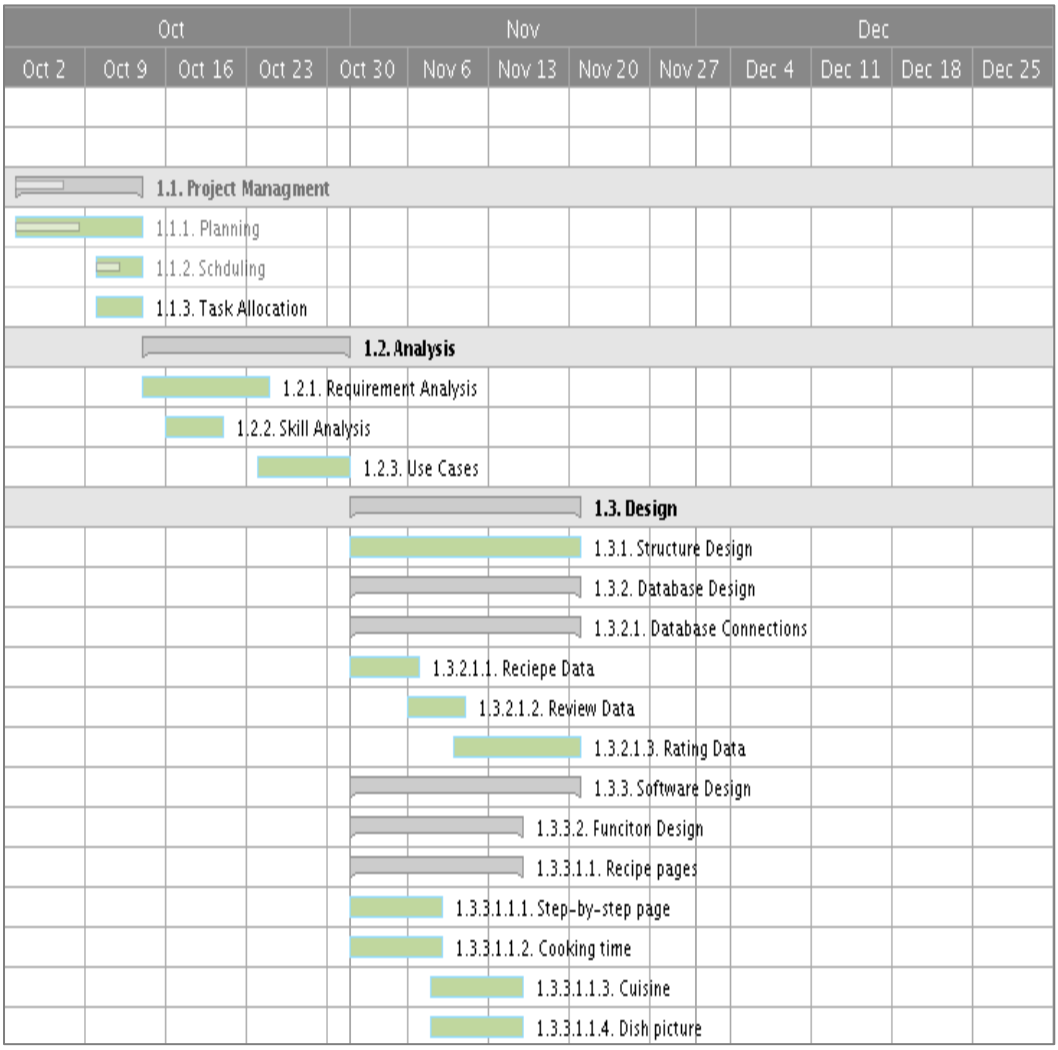
136.25 / 20 = 6.8125 ~ 7 man.week

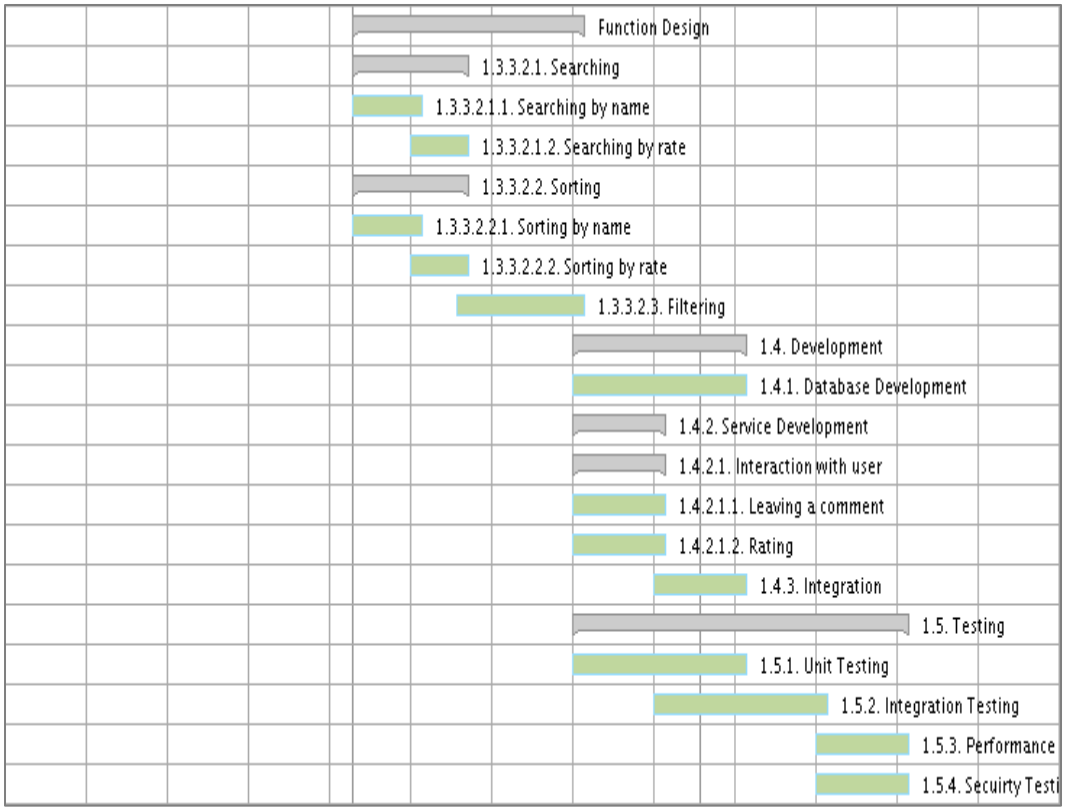
This means that it will take almost 7 weeks for a programmer to complete the program.

4. Resources

Member Name	Member Role	Tasks
Ali Osman ATİK	Project Manager	Service Development Documentation
Baran KAYA	Resource Acquirer / Tester	Testing Analysis
İbrahim TÜRKMEN	Designer	Software Design Structure Design
Caner IŞIK	Database Manager	Database Design Database Development
Emre YENİAY	Analyzer	Analysis Development

5. Schedule





6. Risks

6.1 Risk Analysis Table

#	Risk Definition	Category	Probability (%)	Impact (days)	Risk Exposure
A	A competitor with same scope may arise before finishing the project.	Business	10	9	0.9
B	The costumer (lecturer) may request more features to implement.	Costumer	60	5	3
C	The product may be seen as poor quality for end-users.	QA	80	7	5.6
D	A part of code or data may be lost fully or partially.	Technical	25	10	2.5
E	A component in requirement is not sufficient.	Technology	20	14	2.8
F	A group member may be inexperienced for the given task.	Staff	65	3	1.95
G	A group member may miss the deadline.	Staff	40	7	2.8
H	A group member may quit the project.	Staff	10	16	1.6

6.2 Risk Impact / Probability Chart

Impact → ↓ Probability	Tiny (1-3 days)	Small (4-6 days)	Medium (7-9 days)	Large (10-13 days)	Extreme (>13 days)
Rare (0% - 19%)			A		H
Unlikely (20% - 39%)				D	E
Moderate (40% - 59%)			G		
Likely (60% - 79%)	F	B			
Very Likely (80% - 100%)			C		

6.3 RMMM Table

#	**Mitigation**	**Monitoring**	**Management**	**Activity**
A	Choose a good and unique scope.	Investigate the app store for competitors.	Observe competitor and put features where competitor lacks.	Project Scope
B	Interact with the costumer and give more details about project.	Meetings with costumer to learn its demands more clearly.	Include features those the costumer demanded.	Design
C	Handle the quality management process well.	Get feedbacks from users and testing.	Increase performance and redesign GUI	Testing
D	Put files to either GitHub or Google Drive.	Check branches and the files at the weekend.	Check if it is possible to recover data. If not, recreate the data.	Development
E	Analyze requirement components deeply.	Check requirements also at design and implementation phases.	Change the component with the adequate one quickly.	Requirement Analysis
F	Group members should talk about their skills undoubtedly.	Tell the other group members if one struggles.	Get help from another group members or tutorials.	Task Allocation
G	Give proper deadlines for proper tasks for each team member.	A weekly report about the progress by each group member.	Appoint another member who can work on that task.	Scheduling
H	Talk to group members about their problems.	Check if there is a member who remained silent over 2 weeks.	Distribute the tasks between other group members.	Task Allocation