TRANSPORTATION MANAGEMENT OPTIMIZATION

PROJECT PLAN

Course/CRN:	Software Engineering / 13330
Date:	13/10/2017
Team Members:	Ali UÇAR - 941515004 Özgür AKTAŞ - 150160530 Fatih YILMAZ - 150160531 Okan YILDIRIM - 150160537 Hasan Emre ARI - 150160546

TABLE OF CONTENTS

1. INTRODUCTION

In our globalized world, small businesses that do not develop themselves cannot keep up with the current age for too long. Along with the developing technology, now the produced materials reach more people and the product market expands. Competition is growing with the expandation of market, and small businesses that cannot compete are either bankrupt or bought by big companies. For this reason, companies need to be in constant development and change in order to survive.

A company that works with this vision and produces milk, cheese, yogurt and buttermilk is trying to stand up in the market where competition is abundant is demanding web and desktop application together with an algorithm design from us in order to follow its own transportation management system and reduce transportation costs found. The company's requests and decisions taken together with our employees of the project are explained in detail below.

This company transports these products to a number of city and district weekly. The company has a truck capable of cooling. This truck leaves products at determined centers (pool points) for each city and district. Then, the other local marketing companies distribute to markets and delicatessens from centers. When the company leave extra products on centers in order to avoid more transportations, it poses a problem about freshness due to the wasting more time until products come to consumer from the pool points. However, the company would like to service fresh products to consumers. Therefore, the company deliver products according to orders from centers. In this way, the company services more fresh products to customers because there are more deliveries to pool points.

Every pool points may not give an order weekly. Most of time one truck has not a capacity to take all the orders so that truck has to come back manufacturing center and take products again. If it shortens the path, the truck can leave extra products at some pool points on condition that take it later.

Main purpose of this project is programming of shortest path that truck making way for weekly delivery. In this way, company reduces its expenditure and increases customer satisfaction so that gets more profit. At center points (Manufacturing Center and Pool Points), because loading and unloading to the truck with a forklift takes too short time, this action time does not need to be taken into consideration at any operation. It will be assumed that at pool points there always be enough storage space. For delivery of brochures and promotional items, there may be a single truck to achieve transport to all centers. In this Project, managing the orders made from pool points to manufacturing centers and programming the route of the trucks in such a way that the least amount of way passed is carried. The product centers are making their dairy product orders, as large packages that contain a certain quantity and variety.

Each pool points places orders once a week. In that week the product reaches the pooling center. And these centers may not place orders every week. Each pool points should be able to add and delete order statuses and also display status and history.

The manufacturing center should be able to view orders and their statuses, and program the route so that the truck will travel the least amount of time.

This can be done if the process of stacking extra items in one or more product centers, taking them in the next pass and taking them to another center is the way to go. Orders from the week will never be mixed with orders from another week. Additional product stacking to a center can be set up for another center's suitable order and product can be taken to the requesting center from where the stacking takes place on the next occasion.
Connection routes and distances between city and districts (A, B, C, D, E, F ..) will be entered into the system by the manufacturer.

1.1 Scope

- ✓ Our main goal is finding the shortest path.
- ✓ We aim to service on time (in 7 days).
- ✓ We consider the length of ways.
- ✓ Customer can “sign in” and “login” on website and give order and regulate them by deleting, adding and updating.
- ✓ Customer can see all orders saved on database.
- ✓ Company owners can control the customer’s orders on desktop application
- ✓ Company gives the optimized route to truck driver.

On the other hand, we do not consider;

- ✓ Capacity of pool points
- ✓ Expenditure of driver
- ✓ Loading – unloading time for truck
- ✓ Direct distribution to costumers from pool points

1.2 Deliverables

Deliverable Number	Deliverable	Delivery Date
1	Project Plan	13/10/2017
1.1	Scope of Project	13/10/2017
1.2	WBS of Project	13/10/2017
2	Requirements Specification	27/10/2017
3	Application	10/11/2017
3.1	Database Report	10/11/2017
3.2	Web Report	10/11/2017
3.3	Desktop Report	10/11/2017
4	Interim Report	24/11/2017
5	Test Report	11/12/2017
5	Source Code	18/12/2017

1.3 Epics

Epic Number	Epics	Delivery Date
1	Optimization Algorithm	04/12/17
1.1	Analysis	30/10/17
1.2	Algorithm for shortest path	06/11/17
1.3	Optimization of the algorithm for shortest path with one truck	13/11/17
1.4	Implementation of clustering pull points	27/11/17
1.5	General optimization of algorithm	04/12/17
2	Web Application	04/12/17
2.1	Analysis	23/10/17
2.2	GUI design	06/11/17
2.3	Functionality of Application	13/11/17
2.4	Database connection	27/11/17
3	Desktop Application	04/12/17
3.1	Analysis	23/10/17
3.2	GUI Design	06/11/17
3.3	Functionality of Application	20/10/17
3.4	Database connection	27/11/17
3.5	Embedding algorithm to the application	27/11/17
4	Database	04/12/17
4.1	Analysis	23/10/17
4.2	ER Diagrams	30/10/17
4.3	Constraints	20/11/17
4.4	Data Object Layer	27/11/17
5	Test	04/12/17
5.1	Tests for each sprint	18/12/17
5.2	Test for each module	11/12/17
5.3	System Tests	18/12/17

1.4 Non-Functional Issues

Performance requirements

Usability

Using the system should be used easily. Learning the systems usage by the users should take time less than 1 day.

Security:

The system should be accessible and usable only in authorized ways by authorized users.

The system of giving or editing order should be closed by the company.

The system should keep data entries in check, inconsistent data entries should not be allowed.

Before the calculation of finding optimum transportation path, the system sholud check the related data of pooling point.

Re-usability

The Logistic algorithm of the application should be used for other projects as well.

Performance

Response time of the all system should be fast as possible.

If the process of finding shortest transportation path take time because of huge amount of pool points and alternatives, the user shall understand with a response that the process is going on.

After the process of finding shortest transportation path, if result of the process is accepted, the users should not run the process again.

2. PROJECT PLAN

1. Transport Management Optimization
    - 1.1. Optimization Algorithm **(XL)**
        - 1.1.1. Analysis project **(M)**
        - 1.1.2. Algorithm for Shortest Path**(S)**
        - 1.1.3. Optimization of the Algorithm for Shortest Path with one Truck **(M)**
        - 1.1.4. Implementation of Clustering Pool Points**(L)**
        - 1.1.5. General Optimization of Algorithm**(M)**
    - 1.2. Web Application **(L)**
        - 1.2.1. GUI Design **(M)**
            - 1.2.1.1. Sign Up **(T)**
            - 1.2.1.2. Login **(T)**
            - 1.2.1.3. Ordering **(S)**
            - 1.2.1.4. Ordering History **(S)**
        - 1.2.2. Functionality of Application**(M)**
            - 1.2.2.1. Sign Up **(T)**
            - 1.2.2.2. Login **(S)**
            - 1.2.2.3. Ordering **(M)**
            - 1.2.2.4. Ordering History **(S)**
        - 1.2.3. Database Connection **(M)**
            - 1.2.3.1. Insert/Update/Delete for User Accounts **(S)**
            - 1.2.3.2. Insert/Update/Delete for Order **(S)**
            - 1.2.3.3. Viewing Ordering History **(T)**
    - 1.3. Desktop Application **(L)**
        - 1.3.1. GUI Design **(M)**
            - 1.3.1.1. Sign Up **(T)**
            - 1.3.1.2. Login **(T)**
            - 1.3.1.3. Ordering **(S)**
            - 1.3.1.4. Ordering History **(S)**
        - 1.3.2. Functionality of Application **(M)**
            - 1.3.2.1. Sign Up **(T)**
            - 1.3.2.2. Login **(S)**
            - 1.3.2.3. Ordering **(M)**
            - 1.3.2.4. Ordering History **(S)**
        - 1.3.3. Database Connection **(M)**

1.3.3.1. Insert/Update/Delete for User Accounts**(S)**

1.3.3.2. Insert/Update/Delete for Order **(S)**

1.3.3.3. Viewing Ordering History**(T)**

1.4. Database **(L)**

1.4.1. Analysis **(M)**

1.4.2. ER Diagrams **(T)**

1.4.3. Creating Tables **(M)**

1.4.3.1. Customer **(S)**

1.4.3.2. Products **(S)**

1.4.3.3. Pool Points **(S)**

1.4.3.4. Orders **(S)**

1.4.3.5. Transportation**(S)**

1.4.4. Constraints **(S)**

1.4.5. Data Object Layer**(M)**

1.4.6. Normalization**(S)**

1.5. Test **(L)**

1.5.1. Test for each sprint **(S)**

1.5.2. Test for each module **(M)**

1.5.3. System Test **(M)**

3. ESTIMATES

3.1. External Input(EI)

- (3) Pool point list
- (6) General search system(in pool points)
- (4) User intersection screen(sending for order)
- (3) Province list
- (3) Truck list
- (3) Product list

3.2. External Output(EO)

- (4) Order information screen
- (4) Order history screen
- (5) Products screen
- (7) Customer screen (Sign up, login etc.)
- (4) Result screen
- (4) User interaction

3.3. External Query(EQ)

- (4) Search by user name/order date/company/product
- (6) Sort/Filter result of order
- (3) Search pool points
- (6) Search all of order
- (3) Customer list

3.4. Internal Logical File(ILF)

- (5) Customer File
- (5) Product File

- (5) Pool Point File

Type of Component	Complexity of Components			
	Low	Average	High	Total
EI	3*4	1*4	1*6	22
EO	4*4	1*5	1*7	28
EQ	2*3	1*4	2*6	22
ILF	3*5			15
Unadjusted Function Points : 87				

GSC	Value(0-5)
Data communications	5
Distributed data processing	3
Performance	4
Heavily used configuration:	3
Transaction rate	4
On-Line data entry	5
End-user efficiency	3
On-Line update	5
Complex processing	3

Reusability	3
Installation ease	3
Operational ease	4
Multiple sites	2
Facilitate change	3
	Total = 50

VAF = Total GSC* 0.01 + 0.65

VAF= 50*0.01+0.65=1,15

Adjusted FP= Unadjusted FP * VAF

Adjusted FP=87*1.15= 100,05

Assume the program has a standard rate of 45 LOC/FP and this rate calculated from C#, Sql language and HTML averages. Then,

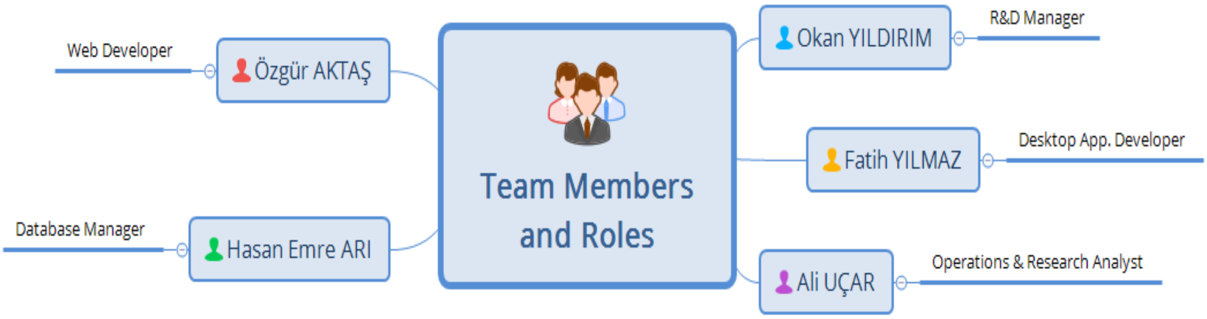
100,05*45=4502 line of code

100/5=20

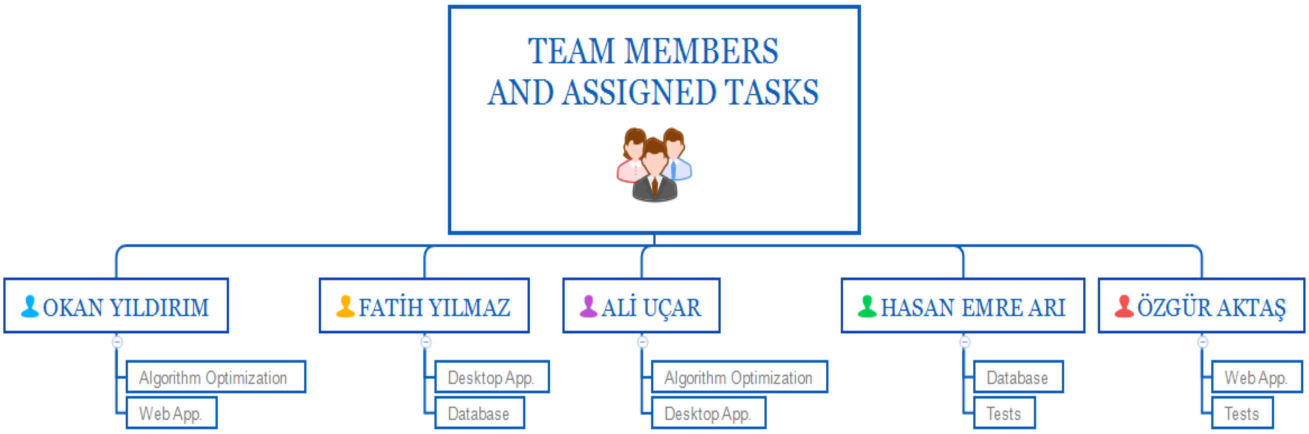
This means that every programmer work 20 days to complete the whole program.

4. RESOURCES

4.1 Team Members and Roles

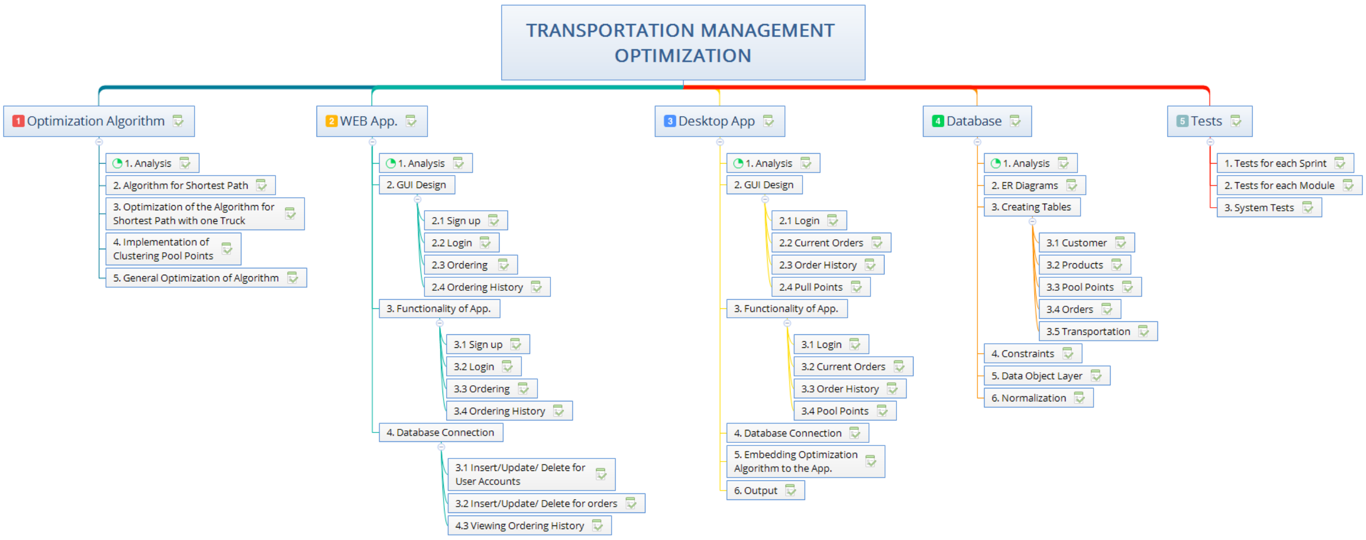


4.2 Team Members and Assigned Tasks

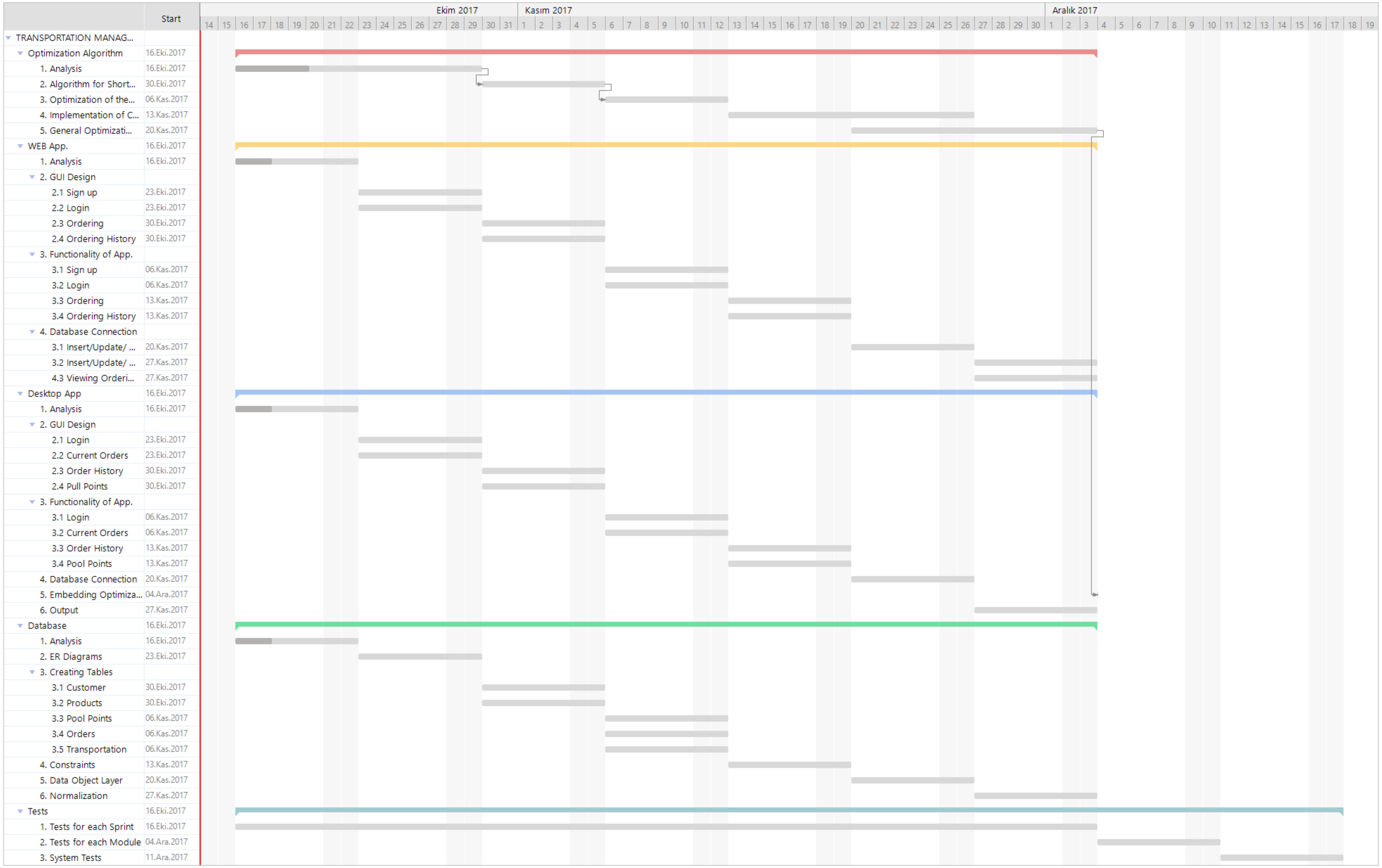


5. SCHEDULE

5.1 WBS Table



5.2 Gantt Chart



	Start
TRANSPORTATION MANAG...	
Optimization Algorithm	16.Eki.2017
1. Analysis	16.Eki.2017
2. Algorithm for Short...	30.Eki.2017
3. Optimization of the...	06.Kas.2017
4. Implementation of C...	13.Kas.2017
5. General Optimizati...	20.Kas.2017
WEB App.	16.Eki.2017
1. Analysis	16.Eki.2017
2. GUI Design	
2.1 Sign up	23.Eki.2017
2.2 Login	23.Eki.2017
2.3 Ordering	30.Eki.2017
2.4 Ordering History	30.Eki.2017
3. Functionality of App.	
3.1 Sign up	06.Kas.2017
3.2 Login	06.Kas.2017
3.3 Ordering	13.Kas.2017
3.4 Ordering History	13.Kas.2017
4. Database Connection	
3.1 Insert/Update/ ...	20.Kas.2017
3.2 Insert/Update/ ...	27.Kas.2017
4.3 Viewing Orderi...	27.Kas.2017
Desktop App	16.Eki.2017
1. Analysis	16.Eki.2017
2. GUI Design	
2.1 Login	23.Eki.2017
2.2 Current Orders	23.Eki.2017
2.3 Order History	30.Eki.2017
2.4 Pull Points	30.Eki.2017
3. Functionality of App.	
3.1 Login	06.Kas.2017
3.2 Current Orders	06.Kas.2017
3.3 Order History	13.Kas.2017
3.4 Pool Points	13.Kas.2017
4. Database Connection	20.Kas.2017
5. Embedding Optimiza...	04.Ara.2017
6. Output	27.Kas.2017
Database	16.Eki.2017
1. Analysis	16.Eki.2017
2. ER Diagrams	23.Eki.2017
3. Creating Tables	
3.1 Customer	30.Eki.2017
3.2 Products	30.Eki.2017
3.3 Pool Points	06.Kas.2017
3.4 Orders	06.Kas.2017
3.5 Transportation	06.Kas.2017
4. Constraints	13.Kas.2017
5. Data Object Layer	20.Kas.2017
6. Normalization	27.Kas.2017
Tests	16.Eki.2017
1. Tests for each Sprint	16.Eki.2017
2. Tests for each Module	04.Ara.2017
3. System Tests	11.Ara.2017

6. RISKS

There is a rule known as Pareto Rule (80-20 rule. It is suggested as %80 of the Italy’s lands are owned by the %20 percent of the population. Then it applied lots of business works. In software projects it is being applied too. Roughly the %20 percent of the whole job meet %80 percent of the customer needs. In this perspective we determine our main risks and their impacts. Our detailed list of risks and its risk table are shown below.

- ✓ Processing Power Limits
  Processing power of finding shortest path can be not affordable.
- ✓ Server load capacity
  Accessing from multiple devices to a single database may overloads the server.
- ✓ Run time estimation
  Execution time of the algorithm may be take more time than planned.
- ✓ Overcoming habits.
  People may don't want to switch to the new system.
- ✓ Team member’s health conditions.
  Team members could be sick.
- ✓ Cooperation of team members.
  Team members may not cooperate with each other.
- ✓ Assignment of works.
  Staff could be assigned to jobs inappropriately.
- ✓ Market’s situation.
  Another similar solution may launch to the market before our project.
- ✓ Multiple device support.
  While developing software, software tools may not show the whole devices previews.
- ✓ Software technologies integration.
  Software technologies may cooperate with each other inefficiently.
- ✓ Change in customer needs.
  Customer may change his/her idea. This may lead a change in whole requirements.
- ✓ Structure dependency.
  Small changes in the requirements may cause huge impacts on the other structure stages.
- ✓ Time estimation.
  Time estimation may be misjudged.
- ✓ Preliminary research.
  Research of required technologies may be misjudged.
- ✓ Work hour’s estimation.
  Assigned man-months may be underestimated.
- ✓ Size of the work packets may underestimated.

6.1 Risk probability / Impact Chart

Risk	Category	Probability	Impact
Change in customer needs.	RE	70%	3
Delay because of structural dependency.	RE	65%	3
Wrong time estimation.	ES	55%	2
Overcoming customer habits.	ST	50%	3
Work hour’s estimation incorrectly.	ES	45%	2
Team member’s bad health conditions.	ST	40%	2
Cooperation of team members.	ST	40%	3
Server load capacity	TE	30%	2
Possibility of launching similar product.	ST	30%	2
Tools insufficiency for multiple devices	TO	30%	1
Software technologies integration.	TO	30%	2
Wrong preliminary research estimation.	ES	30%	1
Wrong run time estimation	TE	20%	1
Correctly Assignment of works.	OR	20%	3
Processing Power Limits	TE	10%	3

TE: Technology ST: Staff OR: Organizational ES: Estimation TO: Tools RE: Requirements

Risk	Affect	Overcoming Method
Change in customer needs.	Project	There must be gradually customer meets for overcoming to go through on a wrong way.
Delay because of structural dependency.	Product	System design should be in a way that the change in customer needs affects the whole system minimally.
Wrong time estimation.	Project	All little stages of the project must be detailly discussed with the all group members
Overcoming customer habits.	Product	Good UI design may increase interest to the software
Work hour’s estimation incorrectly.	Project	Team members should give correct information about their schedule.
Team member’s bad health conditions.	Business	While developing the project team members should be more careful with their health.
Cooperation of team members.	Project	The leader should take some precautions with non-cooperative members.
Server load capacity	Project	Higher technology could be used.
Possibility of launching similar product.	Business	Market should be observed well.

Tools insufficiency for multiple devices	Project	Other technologies should be keeping in mind.
Software technologies integration.	Project	Same technology partners could be chosen.
Wrong preliminary research estimation.	Project	More customer meets can overcome this risk.
Wrong run time estimation	Project	Algorithm optimization can overcome.
Correctly Assignment of works.	Product	All members should clarify their abilities.
Processing Power Limits	Project	Optimization of algorithm will overcome hardware dependency.