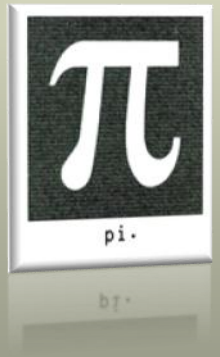
Software Requirements Specification Report

SmARt Shopping Project

Sponsored by ASELSAN

V2. , 2010



Başak Meral

Arda Taşçı

Itır Önal

Deniz Karatay

TABLE OF CONTENTS

1. Introduction

This Software Requirements Specification provides a complete description of SmARt Shopping Project sponsored by ASELSAN, as a final design project of Middle East Technical University Computer Engineering Department, including all the functions and specifications. This document will refer to *functionality*, that is what the resulting application is supposed to do, *external interfaces* which interacts the users, *performance*, *attributes*, that is if the application is portable, or maintainable, and *design constraints* imposed on the implementation such as implementation language, input specifications and output expectations.

1.1. Problem Definition

In order to define the problem, one should imagine a scenario in which a standard customer enters to a supermarket. When a customer enters to a supermarket, he finds himself in a huge crowded environment that has lots of rayon and messy amount of products. For a customer to achieve his purpose he should make a lot of effort at the same time such as finding the exact places of the products he is searching for, while he is trying to carry the products that he is willing to buy and pay attention to his surroundings such as other people. Our team intends to solve this problem which is making shopping easier for an ordinary customer by improving the interaction between the customer and his environment. In other words, the real world problem which this project aims to handle is to collect information from the surrounding area without any effort by a standard user, namely customer in a supermarket. The desired information to be collected is expected to consist of finding the locations of the desired products,

and classifying the products according to their costs and different other categories in huge supermarkets.

1.2. Purpose

This document aims to specify the requirements for SmARt Shopping project that implements augmented reality, and to give detailed information about features of the resulting application, interfaces of the application, and what the application is capable of. This software requirements specification document addresses the ones who are going to develop this system, ASELSAN members and the departmental instructors who are responsible for graduation projects.

It will explain the scenario of the desired project and necessary steps in order to accomplish the task, such as overall description of the project, the definition of the problem that this projects presents a solution, and definitions and abbreviations that are relevant to the project.

One of the main purposes is to establish the basis for agreement between the developers and the suppliers on what this project is to do. The complete description of the functions to be performed by the software specified in this document will assist the potential developers to determine if the software specified in this document meets their needs or how the software should be modified to meet their needs. The other one is to reduce the development effort. The preparation of this SRS will help consider all of the requirements before design begins, and reduce later redesign, recoding, and retesting. The review of SRS can reveal omissions, misunderstandings, and inconsistencies early in the development cycle when these problems are easier to correct.

1.3. Scope

The project which is going to be presented in this document is called SmARt Shopping Application. This application is planned to be used by any person who is willing to do shopping in huge supermarkets. It is designed to run on a hand-held mobile device connected to a head-mounted display. It will allow customers to improve their quality of shopping , and make it easier for them. The promised functionalities of the application will be implemented by means of object recognition, object tracking and using augmented reality tools. By the application, the real world scene captured in the supermarkets will be enriched by the information obtained from the real-time inputs which allow customers to do an interactive shopping.

The application will start to run by the desire of the customer. First, the customer using this application is expected to choose from the products which are defined and listed previously in the application database. After the customer chooses the demands and lists them, the application interface will change into a real-time camera which is capable of recognizing the interest objects such as rayon labels including the desired products, products themselves, or employees of the market. The interact with the user will be maintained by labeling the moving or non-moving recognized interest objects in real-time. Since the user will use a head-mounted display, he will be able to combine the real world with the enriching data without any distraction. The application will also provide information about the found products' cost and quality. The interaction will continue until the user quits application.

Labeling objects will be both according to the chosen list and the previously specified features of the identified objects. A rayon, which has products on sale, is also labelled when it's recognized even if the customer does not add any products to the list from that rayon. The rayons that include the chosen products will be labelled different than the ones who has sale, which can be understood from the features of the rayons.

In order to maintain the accuracy of the application, some assumptions will be made such as tags that will help us to recognize the rayon labels, the shapes of the products, and some color and tag assumptions of products will be made in order to give information about cost or quality of the products .

1.4. User and Literature Survey

Although there exists some applications which combine the augmented reality concepts and image processing, the project which will be explained in this report has unique qualifications. Other applications which combine augmented reality and image processing are mostly designed for entertainment market. As an example of it , an iPhone application which recognizes the foot in the camera, puts a ball image on the screen, in front of the foot. As the player hits the ball image, this ball moves forward and the foot is recognized at real time. As stated before, this application is just for entertainment and does not solve any real world problem.

The completed projects which have the aim to find a solution to a real world problem, don't only use image processing and augmented reality. Those projects use GPS information, and pre-defined attributes about locations. For example, there exists a project called Car Find which uses GPS information of the car and the driver. When the driver is away from his car, he runs the application and the application adds some pointers to the direction of the car, and helps the driver to find the car. So this project gets support from other applications and is not just a combination of augmented reality and image processing.

The previous projects which are also a solution to a real world problem mostly rely on the single images, not on the videos, so they are not real time applications. Related to shopping, there exists a project which recognizes the furniture patterns and puts them on the image of room for the customer to see whether it's suitable for their room or not. Although this project is related to

shopping and helps the customer, it doesn't include augmented reality, instead, it uses the taken image of the room and then puts the recognized furniture on the screen. Hence, this project is not real time.

However, the presented project in this report is a real time project which combines both image processing and augmented reality concepts by an interaction with the whole environment aiming to be a solution to a real world problem.

The potential users of the explained application are everybody who do shopping and wants to find a desired product in an efficient way without losing much time and without paying lots of attention.

1.5. Definitions and Abbreviations

SRS: Software Requirements Specification

AR: Augmented Reality , a term for a live direct or indirect view of a physical real world environment whose elements are augmented by virtual computer generated sensory input such as sound or graphics.

OpenCV: Open Source Computer Vision Library

IDE: Integrated Development Environment

GPS: Global Positioning System, is a space-based global navigation satellite system that provides reliable location and time information in all weather and at all times and anywhere on or near the Earth when and where there is an unobstructed line of sight to four or more GPS satellites.

Head-Mounted Display: a display device, worn on the head or as part of a helmet, that has a small display optic in front of one or each eye.

ER: Entity – Relationship Diagram, a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to

represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

GUI: A graphical user interface, a type of user interface that allows users to interact with programs in more ways than typing such as computers.

UML: Unified Modeling Language, a standardized general-purpose modeling language in the field of software engineering.

1.6. References

[1]	IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
[2]	Fowler, Martin. " UML distilled: a brief guide to the standard object modeling language." Booch Jacobson Rumbaugh, 2004
[3]	Bruade, Eric J. "Software Engineering: An Object-Oriented Perspective." Wiley, 2001.

1.7. Overview

This document contains a detailed description about the project SmARt Shopping. In the introduction part, it mostly gives a general overview about the project including the definition of a real world problem that project intends to solve, the scope of this project, and the information about similar projects and how they differs from this project. Also in this part, the purpose of the SRS and the scope of the project are explained. Moreover, user and literature survey of this part explains the similar projects done before is declared. Second part of the

document is the overall description of the project. This part explains the product perspective of the application, the functions included in the application and the constraints, assumptions and dependencies of the desired application. The specific requirements of the project are explained in the third part of the document. This part includes interface requirements and functional/nonfunctional requirements of the project. In the fourth part of this document, data models and their descriptions are explained in detail with the relationship between them. The behavioral model and its descriptions are mentioned in the fifth part including the state diagrams. Next part includes the team structure and introduction of the team members of the project. The final part is conclusion part which gives a brief summary about the whole document.

2. Overall Description

This section will give information about product perspective, product functions, constraints, assumptions and dependencies.

2.1. Product Perspective

SmARt Shopping Application is totally an independent application which is not related to any other system and not a component of a larger system. This application has only one type of user, and thus there is no functionality differences between users. That's why, the application only has one type of user interface. The interface of the application has a starting menu, which consists of a list of predefined products that the supermarket provides. The interface allows the user to choose from the products. After the choice, the interface changes into a real-time camera scene, and until the user quits the application, the interface remains as a video. This video consist of the real scene enriched by surrounding

information by labeling in terms of augmented reality. The interface quits by the demand of the user.

In terms of hardware, we'll use a portable device with a camera ,display screen and control buttons connected to it. We will be recognizing the data which is captured by the camera and the screen will display that captured video with augmented reality labels.

In terms of software interfaces, this application will run on Windows 7 Operating System, and it will be implemented making use of Eclipse IDE with OpenCV plug-in integrated on it as a mathematical package. OpenCV is chosen as a mathematical package because it supports basic image processing, structural analysis , camera calibration , motion analysis, object recognition, basic GUI , and image labeling . Also OpenCV is a fast library which is appropriote for real-time video input-outputs.

Since our application does not require huge amount of memory, that is the information captured can fit into the memory of the device, we will not need secondary storage devices.

2.2. Product Functionalities

The major functionalities of our application are:

- **Creating Lists from the Products**

The program allows user to choose the products that he wants and adds them to the list or delete the products that he does not want from the list.

- **Object Detection**

The program detects predefined objects(products, rayons, salesmen) according to their shapes and colors.

- **Real-Time Object Tracking**

The program tracks the detected objects in real time even if the orientation of the camera changes.

- **Labeling**

The program can label the detected objects according to their features like wanted, on-sale, high-quality and popular.

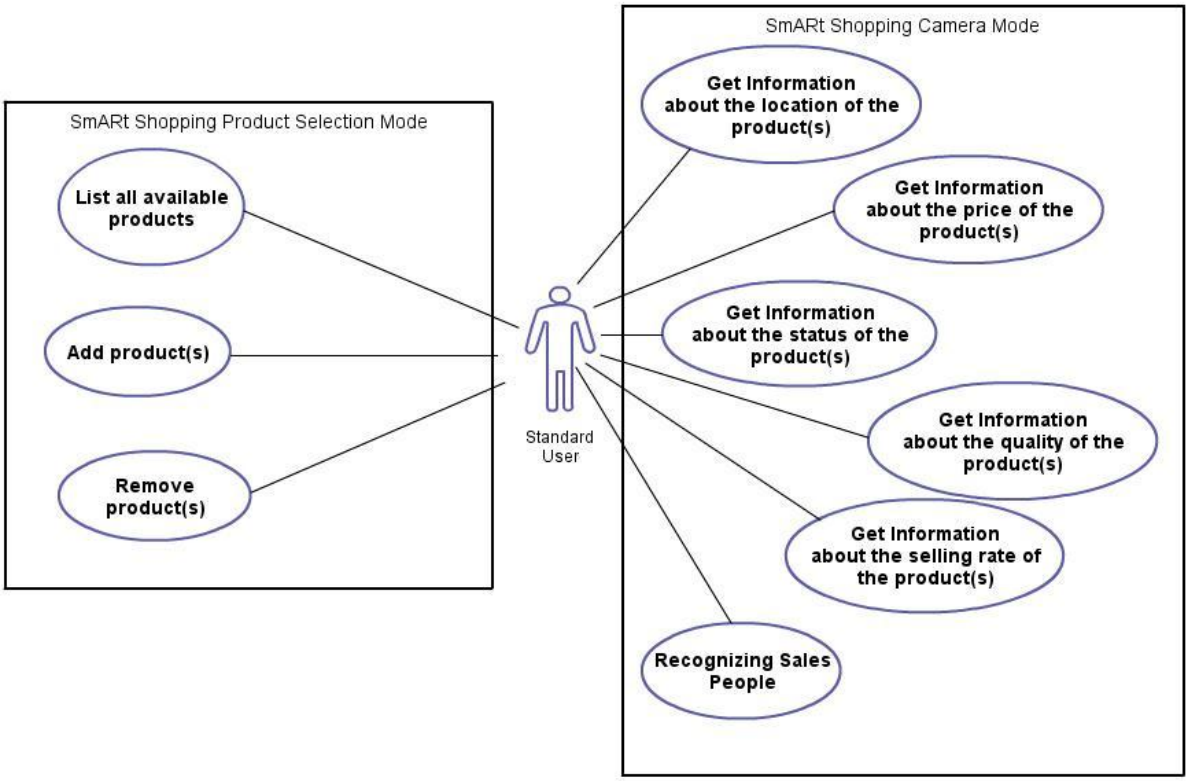


Figure 1 - Use Case Diagram

2.3. Constraints, Assumptions and Dependencies

In order to maintain the accuracy, following assumptions are made:

- Rayon can be recognized up to 5 meters.

- Products can be recognized up to 1 meter.
- Products will be uniquely defined by geometric shape and color couple.
- Different qualifications of the same type of product information will be obtained by letters on them.
- Rayon flags will have the form of square marker patterns.
- Sales person is assumed to wear a unique color that no product has the same color on it.
- Sales person will be recognized up to 3 meters.

The application has to process and answer in real time, but, obviously, this application is limited by the performance of the portable device and the camera. The application does not have any safety and security concerns.

3. Specific Requirements

The specific requirements of this project will be considered in following three subsections, which are interface, functional and non-functional requirements.

3.1. Interface Requirements

As explained before, since there are just one type of user, the application will have only one interface. The interface of the application will be a simple start menu at the beginning and then turn into a real-time camera mode with enriched data. When the user starts the application, this act will call the function *Start()*, and this function will bring the product selection mode on the screen. The product selection mode screen will be divided into two columns, one of the columns will be the predefined product list, and the second column will display

the chosen product list. The user will be able to move around between the columns, left and right, and inside the columns, up and down. The user will be able to call the function *AddProduct()* by choosing an entry in the first column, this function call will result in adding the product into the second column, namely the demand list of the customer. The user will also be able to call the function *DeleteProduct()* by deselecting with the help of control buttons, and this function will result in removing the product entry from the demand list of the customer. When the customer is done choosing the products that he wishes to buy, he will be able to save the list by pressing the save button with the help of control buttons and this act will call the function *Load()*, and move to the camera mode. When the user moves into the camera mode, until he quits the application by means of control buttons calling *Quit()*, every other function calls will be made by the application itself according to the input coming from the environment.

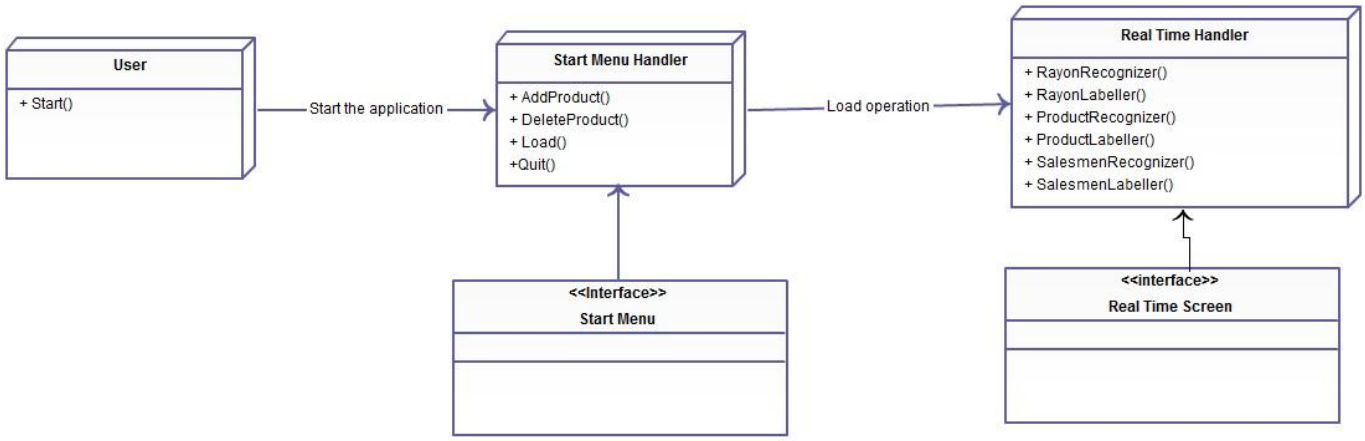


Figure 2 – Interface Requirements

3.2.Functional Requirements

Functional requirements of this application can be categorized in two parts, namely user functional requirement, the functions called with the user activity , and device functional requirements, the functions called automatically.

3.2.1. User Functional Requirements

3.2.1.1. Start application

Description: The user shall start the application.

Assumption: The application is loaded on the device.

How: By running the application.

3.2.2. Product Selection Mode Handler Functional Requirements

3.2.2.1. Add product

Description: The user shall add the product to the list.

Assumption: The list is already defined and displayed on the screen. The screen is on menu mode, it's not displaying the output of the camera yet.

How: The user adds the current product to the list.

3.2.2.2. Delete product

Description: The user shall delete the product from the list

Assumption: The list is already defined and displayed on the screen. The screen is on menu mode, it's not displaying the output of the camera yet.

How: The user deletes the current product from the list.

3.2.2.3. Load List

Description: The user shall load the list to the program's memory

Assumption: The list is already defined and displayed on the screen. The screen is on menu mode, it's not displaying the output of the camera yet.

How: The user saves the list to the program's memory.

3.2.2.4. Quit

Description: The user shall exit from the application whenever he wants

Assumption: There will be no assumption for the system to execute this function.

How: Using ESC button of the keyboard the user exits the program

3.2.2. Real Time Handler Functional Requirements

3.2.2.1. Recognize Rayon

Description: Rayon patterns are recognized according to their square marker patterns which are hung in front of each rayon.

Assumption: Rayon patterns are already defined in program's memory. Each rayon has a unique square marker pattern. Patterns should be closer than 5 meters to be recognized.

How: When the rayon patterns which are already defined are matched with object on the screen, the object is recognized with 80% accuracy.

3.2.2.2. Label Rayon

Description: Rayons are labelled after they are recognized.

Assumption: Rayon patterns are recognized

How: After the rayon patterns are recognized, they are labelled differently according to some features like including products on-sale and including chosen products by the user.

3.2.2.3. Recognize Product

Description: Product patterns are recognized according to their shapes and colors.

Assumption: Product patterns are already defined in program's memory. Objects should be closer than 1 meters. Every product has a unique pattern.

How: When the product patterns which are already defined are matched with object on the screen, the object is recognized with 80% accuracy.

3.2.2.4. Label Product

Description: Product patterns are labelled after they are recognized.

Assumption: Product patterns are recognized.

How: After the product patterns are recognized, they are labelled differently according to their quality, cost and popularity.

3.2.2.5 Recognize Salesmen

Description: Patterns of the salesmen are recognized according to the color of their shirt.

Assumption: All the salesmen will be wearing the same color.

How: When the salesman patterns which are already defined are matched with object on the screen, the object is recognized with 80% accuracy.

3.2.2.6 Label Salesmen

Description: Salesmen are labelled after they are recognized.

Assumption: Salesman patterns are recognized. Objects should be closer than 3 meters.

How: After the salesmen are recognized, they are labelled.

3.3. Non-functional Requirements

3.3.1. Performance requirements

First, this application will be used by a single user. There will be no multiple user handling since the application runs on a single portable device without needing any network. The amount of the input is huge since the input data of the application is the video captured by a single camera. The information among the data will be achieved by reducing input into a set of features of the interested objects, which is also essential in object recognition. Interest objects may be more than one at the same time, that the software has to handle multiple object recognition with a single camera. Also, the objects may be moving, or non-moving, and this fact should not effect the performance of the application. The major issue here is the application should answer in real-time, namely, the recognizing and labeling operations has to be handled in less than 1 second. Also the application should be able to recognize more than %80 of the interested objects simultaneously captured. The assumptions to recognize and track the objects has already been defined in *2.3 Contraints, Assumptions, Dependencies* part of this document.

3.3.2. Design constraints

The SmARt Shopping Application will follow the following design constraints:

Programming language that will be used during the project is C++. Since OpenCV will be used for object recognition parts and augmented reality tools of the project, C++ is the most appropriate programming language that supports OpenCV very rapidly.

The system shall run on a portable device which also supports a camera. Also, the processor speed, the quality of the graphics card should be sufficient enough to be able to run the program.

The portability of the system depends on the portability of the device in which system will be running. Since this will be a real-time application and notebooks will be used in order to maintain the speed, system can be used in anywhere that customer will carry his notebook.

Since the system will run on a single machine and it will not be dependent to a bigger system or any other computers, there will not be any security or reliability problems.

SmARt shopping project is designed to be extensible, that is open to changes. Whenever new patterns are added to the memory of the program, by the help of some additions, the application will handle recognizing them. Moreover, the start up menu is open to additions or removals, if necessary.

4.Data Model and Description

This section will give information about the data objects, their attributes and the complete data model.

4.1. Data Description

This section will give information about the data objects related to this project, the relationship among them, the attributes of the data objects and the complete data model with data objects' functions included.

4.1.1. Data objects

The data objects are Product, Rayon and Sales Person. Product data objects has six attributes called, ProductID, ProductName , ProductCost, ProductSellingRate, ProductQualityRate, ProductStatus. The second data object, Rayon has four attributes, namely RayonID, RayonName, RayonType and RayonStatus. Finally Sales Person data object has only one attribute called PersonID. Since our data is reduced from the input which is a real time scene captured from a camera, sales person does not have many attributes because it is a moving object in order to recognize and label according to the specified rules of the application, and each identified sales person will be treated in the same way.

4.1.2 Relationships

As explained in the previous section, we have basically three data objects which are namely Product, Rayon and SalesPerson. Product and Rayon data objects have a mutual relationship in such a way that rayons include products, and every product belongs to a specific rayon. Other than that, we have sales person as our moving objects, which is responsible for rayons. ER diagram of the data objects are presented in the following section.

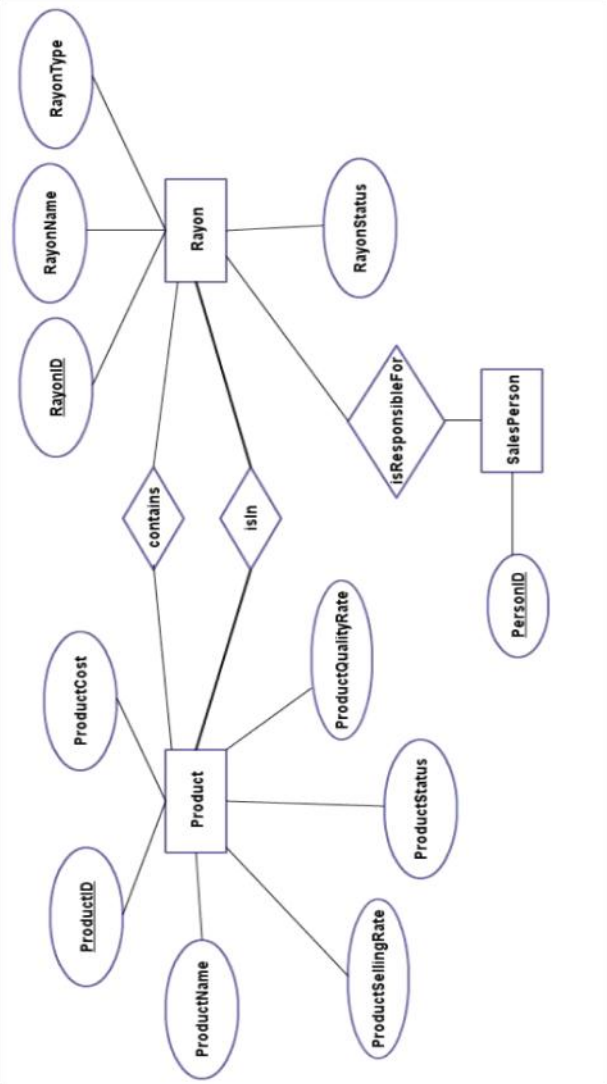


Figure 3 - ER Diagram

4.1.3 Complete Data Model

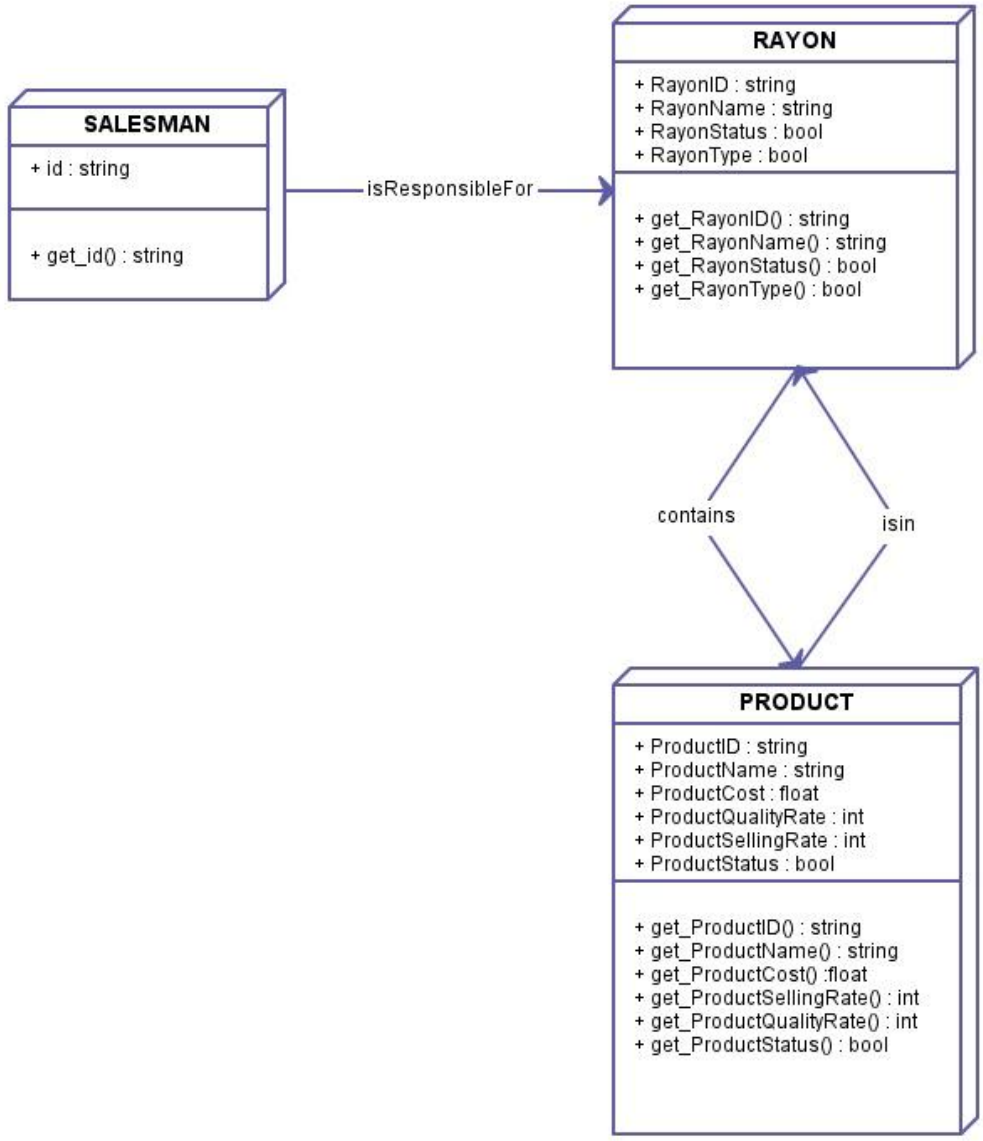


Figure 4 - Class Diagram

4.1.4 Data Dictionary

4.1.4.1. Product

- ProductID : a unique number given to every product.
- ProductName : Name of the product.

- ProductCost:: The cost information of each product.
- ProductSellingRate: The selling rate information of each product.
- ProductQualityRate: The quality rate information of each product.
- ProductStatus: Status of the product, if it's in the list or not.

4.1.4.2. Rayon

- RayonID : a unique number given to every rayon.
- RayonName : Name of the rayon category.
- RayonType : Status of the rayon, if its in sale or not.
- RayonStatus: Status of the rayon, if it's targeted by customer, or not.

4.1.4.3. Sales Person

- PersonID: a unique number given to each Sales Person who are in the scene. There are no other attributes in the sales person class which is unique for each sales person.

5. Behavioral Model and Description

Among the behavioral models, we decided that the best way to represent our project is to use state machines since the application performs a loop operation and does not terminate until the user wants to quit. The product selection mode also performs a loop operation in itself since until the user moves into the camera mode, adding and deleting operations can be done numerously. When moved into the camera mode, since it is a real time application, for every frame that the camera captures the steps will be repeated. *State machines* take

each step based on when events occur in the environment of the behavior being performed. The inputs to each step are calculated as part of the step itself. That's why we'll explain the behaviour of our application in this way.

5.1. Description for Software Behavior

The first state is the start state in which the application starts and invokes enter the product selection mode event. That event leads to the product selection mode state, namely smARt shopping interface. In that state, the user may choose to add an item to the chosen product list or delete an item from the chosen product list. If adding event is chosen, the next step will be add item to the chosen product list state, or if removing event is chosen, the next step will be remove item from the chosen product list state. After those events, if the user wants, he can add another items to the list or delete another items from the list by following the same states. When adding and deleting choices are finished, the list will be loaded with the load event. The following state is the real time screen and at that step, recognizing rayon, recognizing salesman and recognizing product events are handled simultaneously, according to the objects recognized via camera. Each of them are labelled according to their features, when a product is recognized, it moves into product label state, when a salesman is recognized it moves into salesman label state and when a rayon is recognized it moves into rayon label state. After those labeling steps, they continue to recognize the objects by following the same steps. When the user wants the application to go to end state, by the quit event the end state is achieved.

5.2. State Transition Diagrams

The state transition diagram shown below explains the overall behaviour of the system:

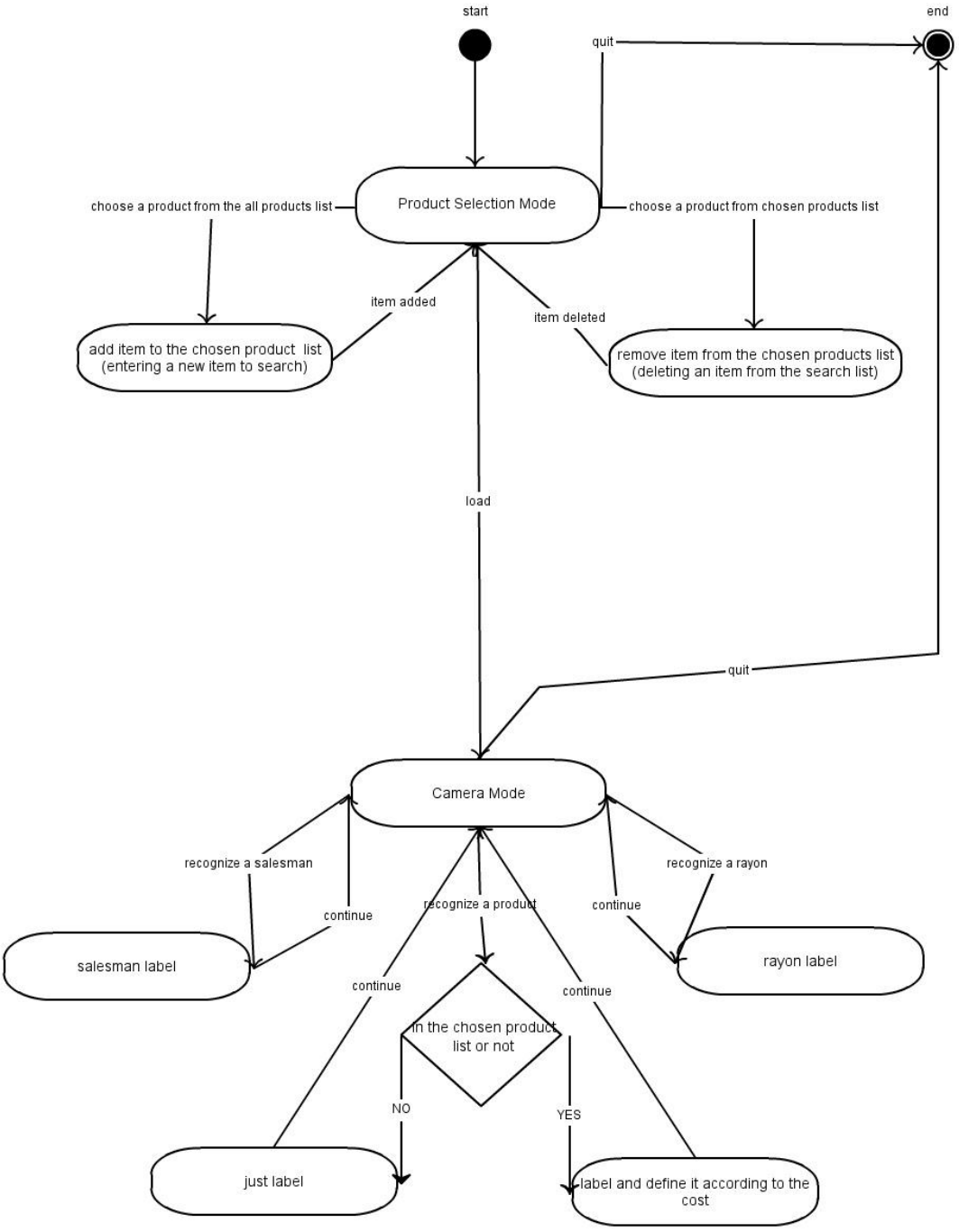


Figure 5 - State Diagram

6. Planning

In this part of the document, the structure of the team responsible from the project, the basic schedule, and the process model will be presented.

6.1. Team Structure

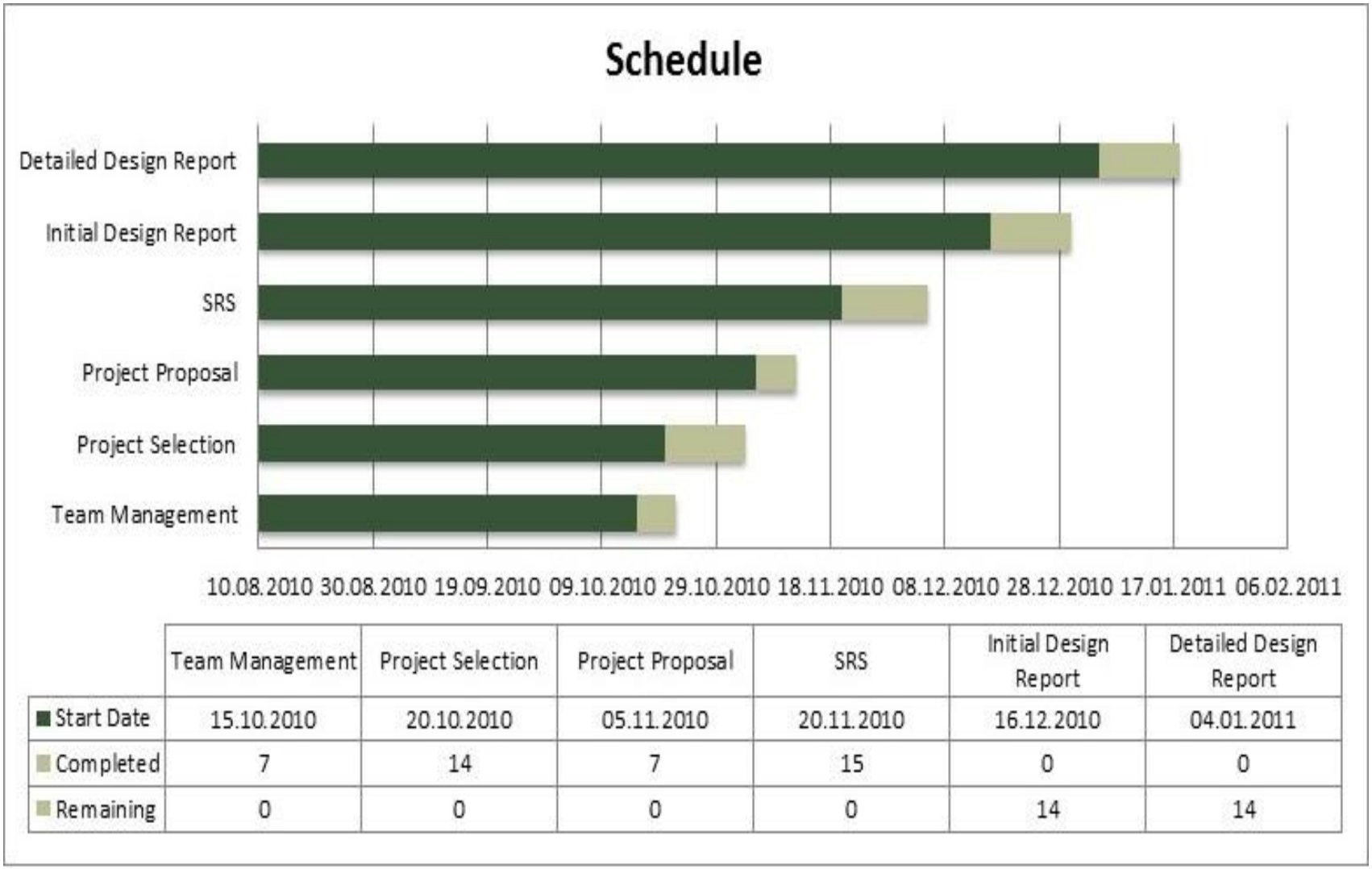
This team's members have known each other for a long time and have experience on division of labor since they have worked together on several projects previously. Every member's opinion is important to each other and the experience and knowledge of developments tools are nearly the same. Thus, our team's structure is decided as "Democratic de-centralized" structure. We plan to divide the workload equally.

Team decided Deniz and Başak as the contact people. Main roles of the members are explained below:

Arda: The harmonizer, The Devil's Advocate
Başak: The encourager, the optimist
Deniz: The initiator, the aggressor
Itır: The gatekeeper, The information seeker

6.2. Estimation (Basic Schedule)

The gantt chart that shows the tentative schedule of the project is as follows with the tasks and dates included:



	Team Management	Project Selection	Project Proposal	SRS	Initial Design Report	Detailed Design Report
Start Date	15.10.2010	20.10.2010	05.11.2010	20.11.2010	16.12.2010	04.01.2011
Completed	7	14	7	15	0	0
Remaining	0	0	0	0	14	14

Figure 6 - Gantt Chart

6.3. Process Model

After examining the process models, our team decided on The Incremental Model. In order to explain this model, one needs to explain The Waterfall Model. The Waterfall Model , sometimes called The Classic Life Cycle suggests a systematic sequential approach to software development that begins with customer specification requirements and progresses through *planning*, *modelling*, *construction*, and *deployment*, *communication.* The Incremental Model combines the elements of the waterfall model applied in iterative fashion. We have chosen this model because the incremental model focuses on the delivery of an operational product with each increment.

7. Conclusion

This software requirements specification document gives information about the project SmARt Shopping which implements augmented reality. First, the real world problem is defined and the solution that our application proposes is explained. This explanations includes basically functionality of the application, interface requirements of the application, performance, attributes,and design constraints imposed on the implementation. In the overall description part, all of the functions that this application will perform is explained one by one. User and function relationships, user roles and characteristics are modeled. The assumptions that will maintain the accuracy are made in order to sustain a reliable application, in addition to that data models and behavioral models are presented. Finally, we presented the team structure and basic expected schedule and the process model of our team.