1. Introduction

Purpose of this project is to provide mobile platform to users, especially people who are responsible with storage management.
This system, or application, will be used to store various products information at the users warehouse. Like, the products location, id, product attributes.

This design document explains detailed specifications of our Inventory Management App (IMA), before the implementation phase. Design of the project is prepared with respect to Requirements Specification Document (RSD) that is arranged in specification Phase.

1.1. Purpose

The objective of this document is introducing a detailed explanation of the designs of Inventory Management App (IMA), developed for the storage owners. First of all, this document is confirmed for implementation phase which helps programmers to understand design clearly. We can see this document as a guideline for them. Also, this document can be used by designer to upgrade or change existing design.

1.2. Scope

This document includes detailed description of the software architecture of the IMA. It is specified version of RSD which is supported by the use cases and their transformed version to the sequence and activity diagrams. Additionally, class diagrams show how program should be implemented.

1.3. Definitions, Acronyms and Abbreviations

IMA - Inventory Management App
AOS – Android Operating System
GUI – Graphical User Interface
UI – User Interface
IDE – is a software application that provides comprehensive facilities to computer programmers for software development
Application – is computer software designed to help the user to perform specific tasks.
Wi-Fi – is a popular technology that allows an electronic device to exchange data via wireless (using radio waves) over a computer network, including high-speed internet connections.
Android – is a Linux-based operating system for mobile devices such as smartphones and tablet computers.
Linux – A Unix-like computer operating system.

Java - A programming language will be used for programming on Android.
AWS - A cloud computing system.
NoSQL - encompasses a wide variety of different database technologies that were developed in response to the demands presented in building modern applications.
API - An application programming interface.

1.4. References

IEEE Standard 1016-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.

1.5. Overview

This document is prepared based on the "IEEE Recommended Practice for Software Design Documentation" which describes how the software design document is composed. In section two, there is a little summary about software requirements document such as assumptions, constraints, system environment and design methodology. For the sections three to five, it is stated architectural and data intended design with diagrams.

2. Design Considerations

2.1. Assumptions

The user group of our system is assumed to have knowledge of using mobile devices such as smart-phones. Additionally, application is compatible with devices which have Android operating system. That's why, users shall be aware of Android Systems. The system is accessible unless there is a Wi-Fi or 3G connection to fetch data from web.

2.2. Constraints

The system will be implemented using Java, and shall need version 4.4 and above Android Operating System.
Back-end of the system will be implemented using nodejs version 8.10 and above.
Database will be noSQL dynamoDB.

2.3 System Environment

Application will be developed on Android Studio. Back-end will be developed on Amazon web services console. For backend code AWS Lambda and for API's AWS API gateway will be used.

Each user shall need an Android compatible mobile device, smart-phones or tablets.

3. Architecture

3.1. System Design

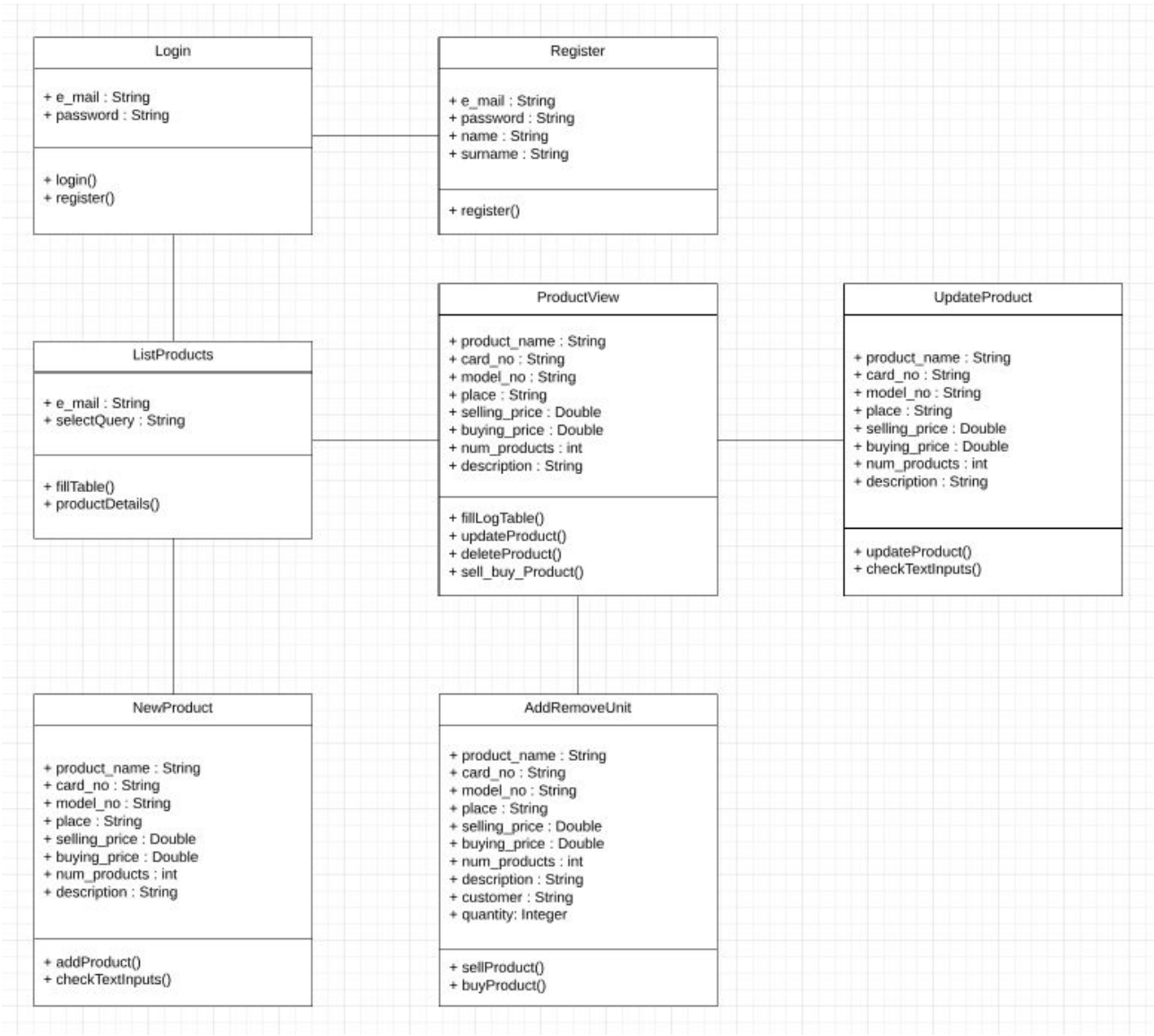
The block diagram below represents the principal operational parts of the system and their interactions.

4. Data Design

Database is chosen to be with noSQL structure. Reason for this is, we are making an inventory management app. Inventory categories of the stored items may not be always fixed. Thus, noSQL will give us more freedom over the data we will be storing on database. In first design we will have three database tables to start with. “Persons” will store user information. “Products” will store product information. “Product_logs” for storing the metadata of the Products that are stored.

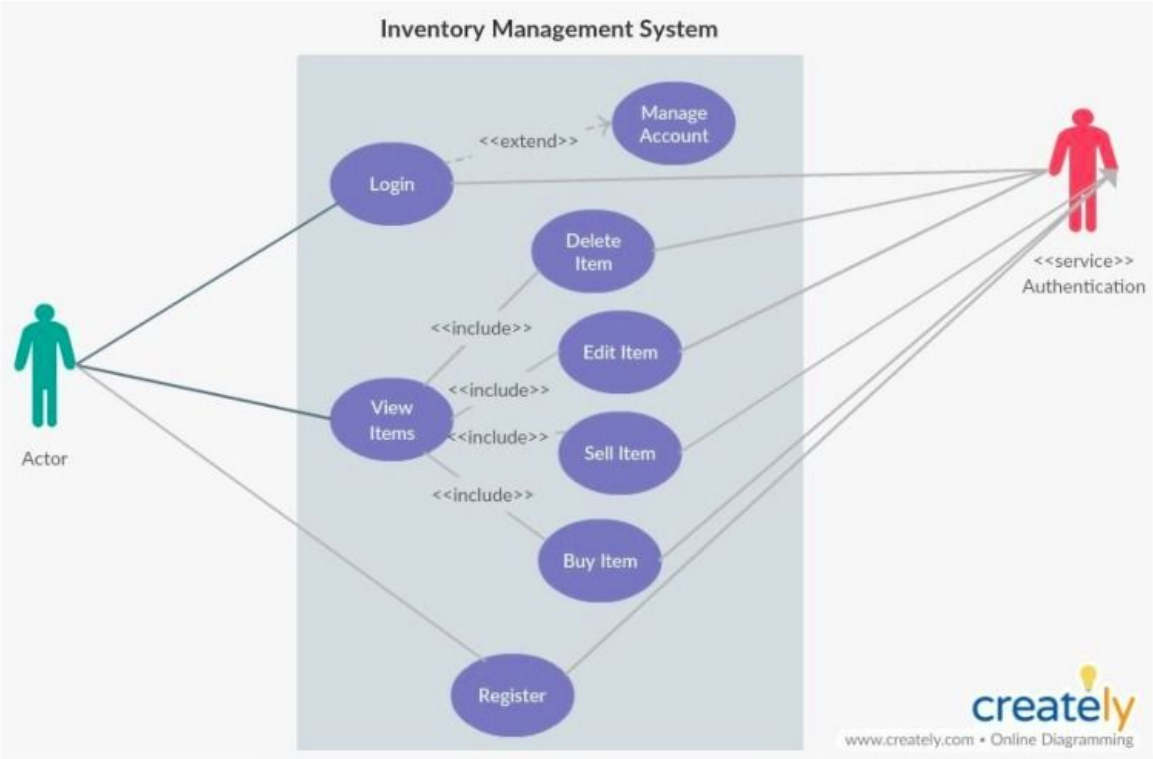
5. Component Design

a) Class Diagram



b) Use Case Diagram

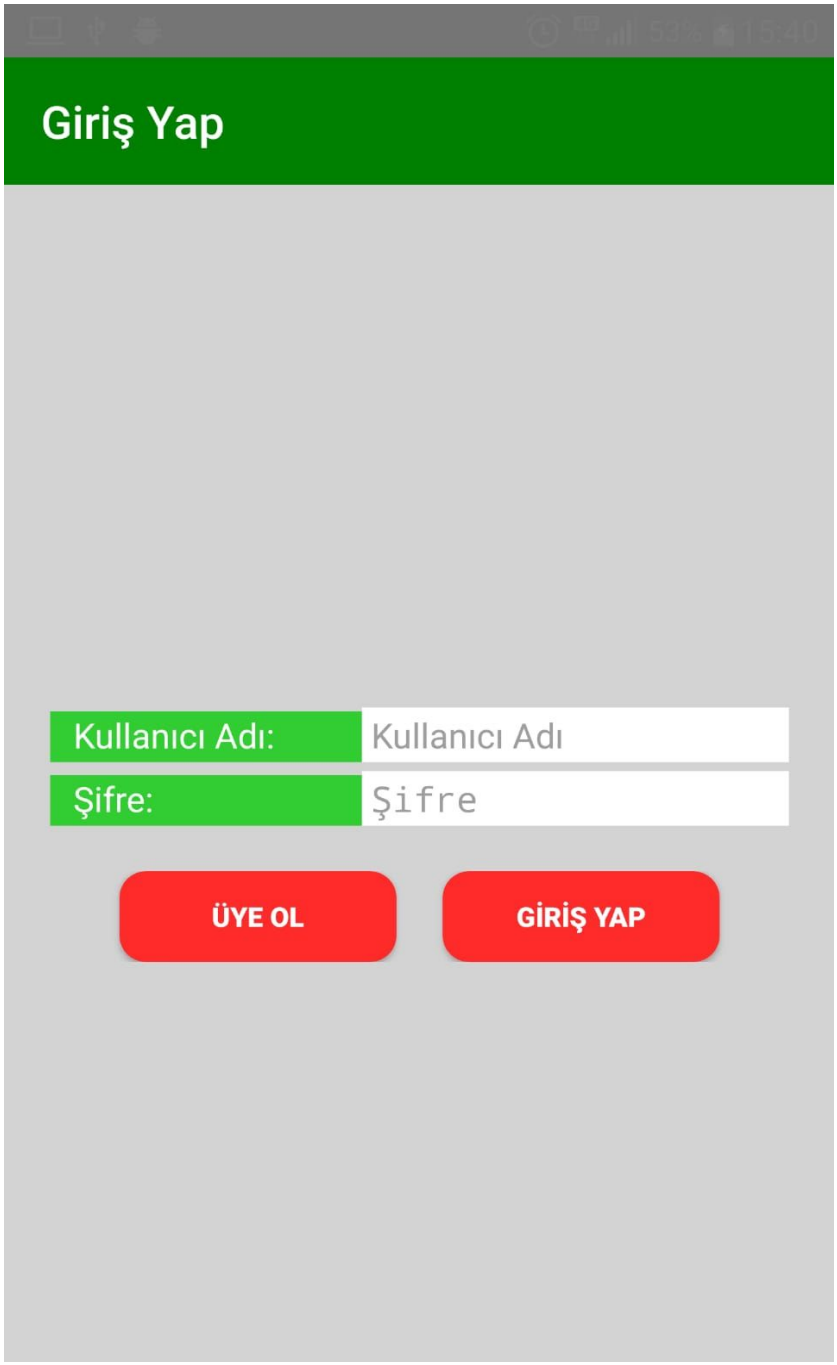
Use Case Diagram:



6. User Interface Design

a) Login Screen:

First screen that will be seen by user. User will enter ID and Password if has an account. Otherwise he/she has to sign up.



b) Product List:

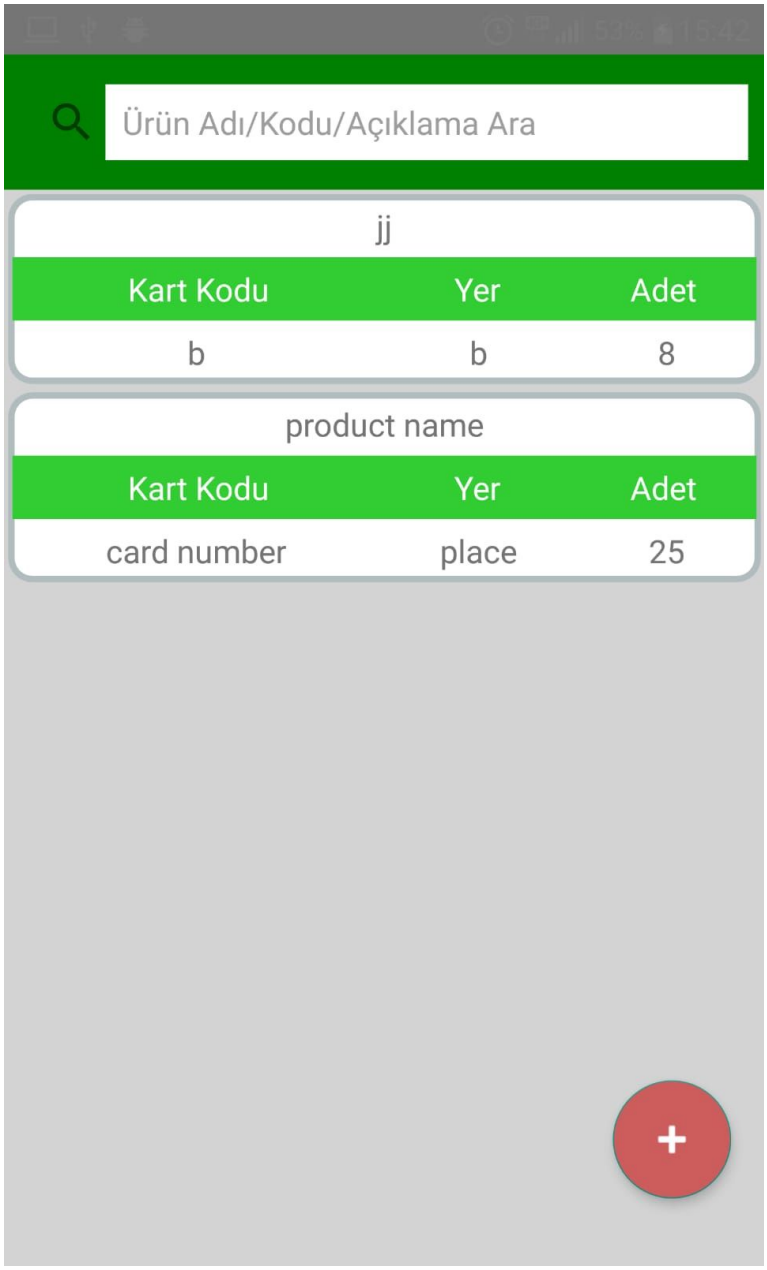
This screen shows recorded products of the user with following informations:

1. Product Name
2. Card no of the product
3. Where Product is
4. Number of products

User will be able to see more details about the product that is clicked.

User can add a new product using “ + “ button at bottom right of the screen.

User can search product using search bar at top of the screen.

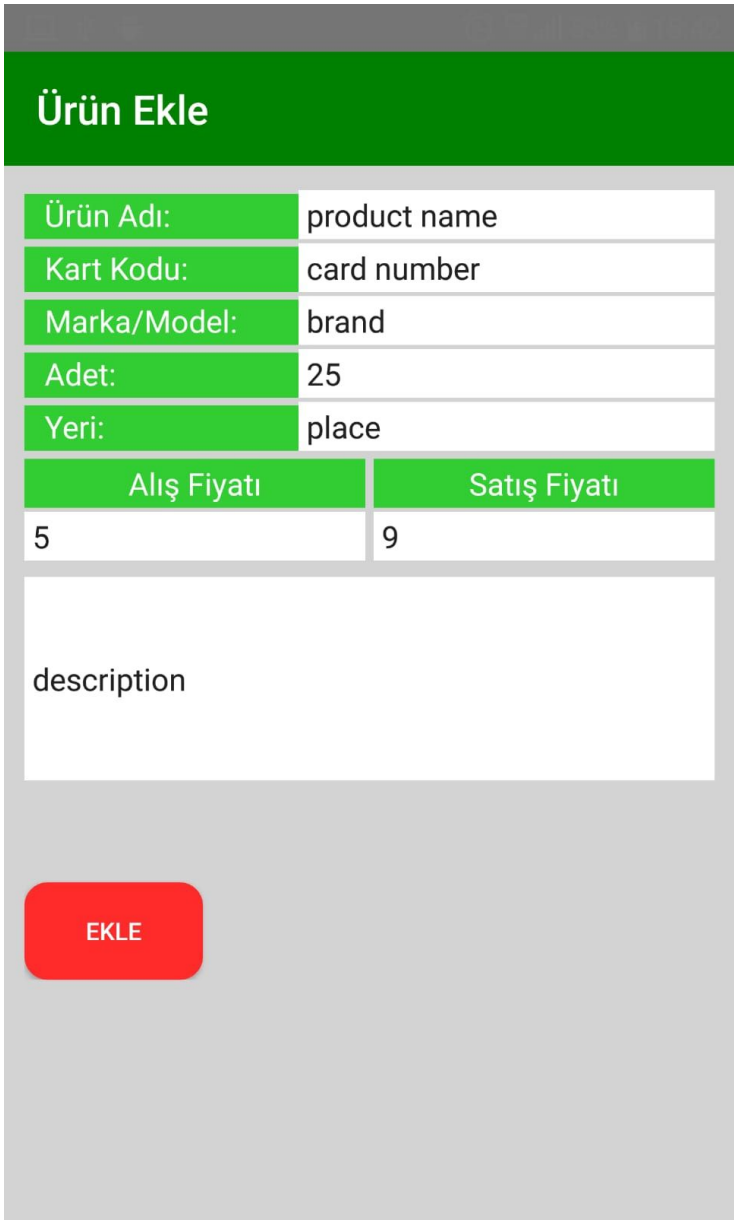


c) Add New Product:

User will be able to add a new product to list using this screen with following informations:

1. Product Name
2. Card No
3. Brand
4. Number of Products
5. Place
6. Selling Price

7. Buying Price
8. Description (Optional)

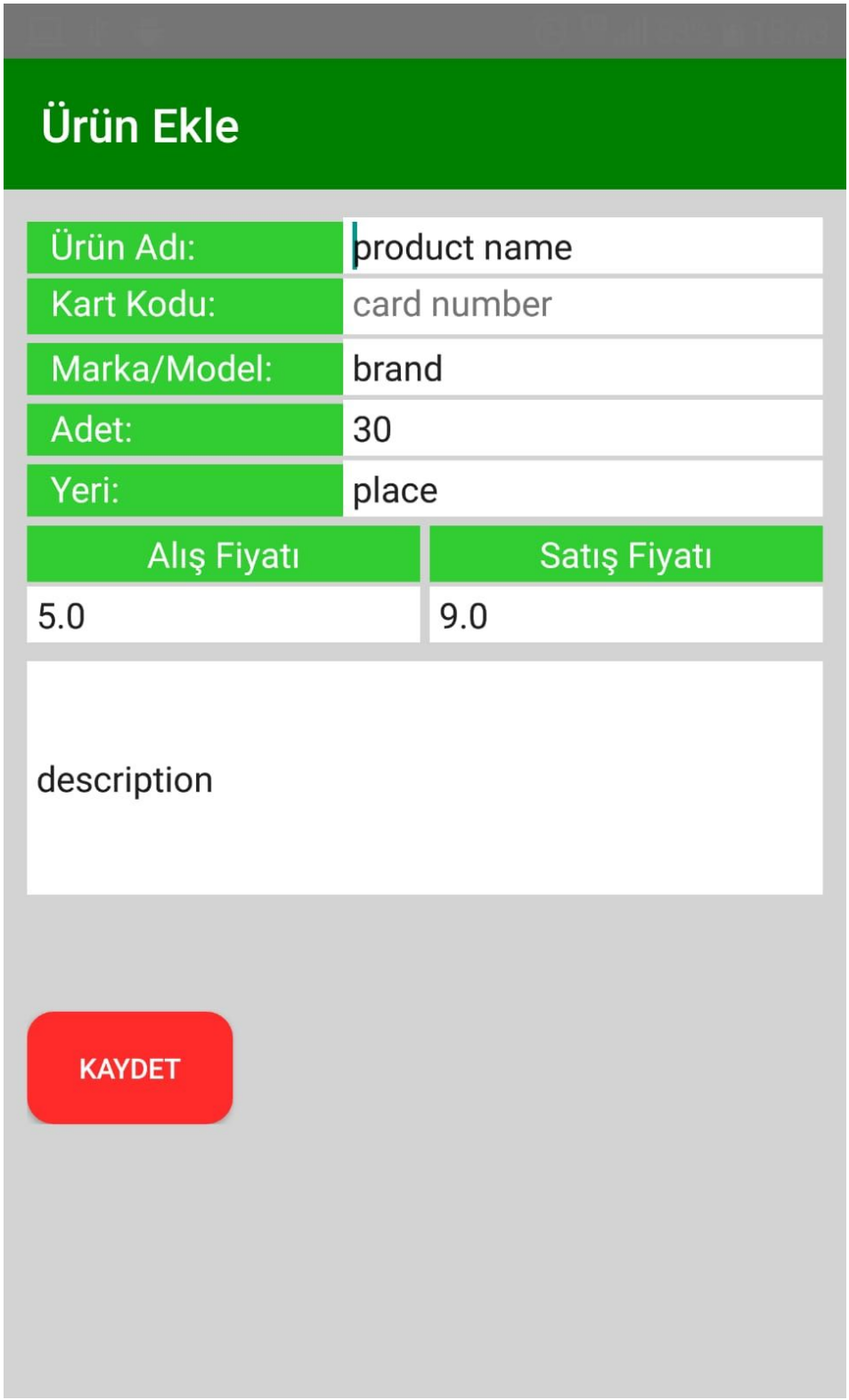


d) Product View:

User can see all informations about the product at this page. Also user will be able to Update / Delete / Sell & Buy product using the buttons. At the bottom list user can see information about sell & buy operations.

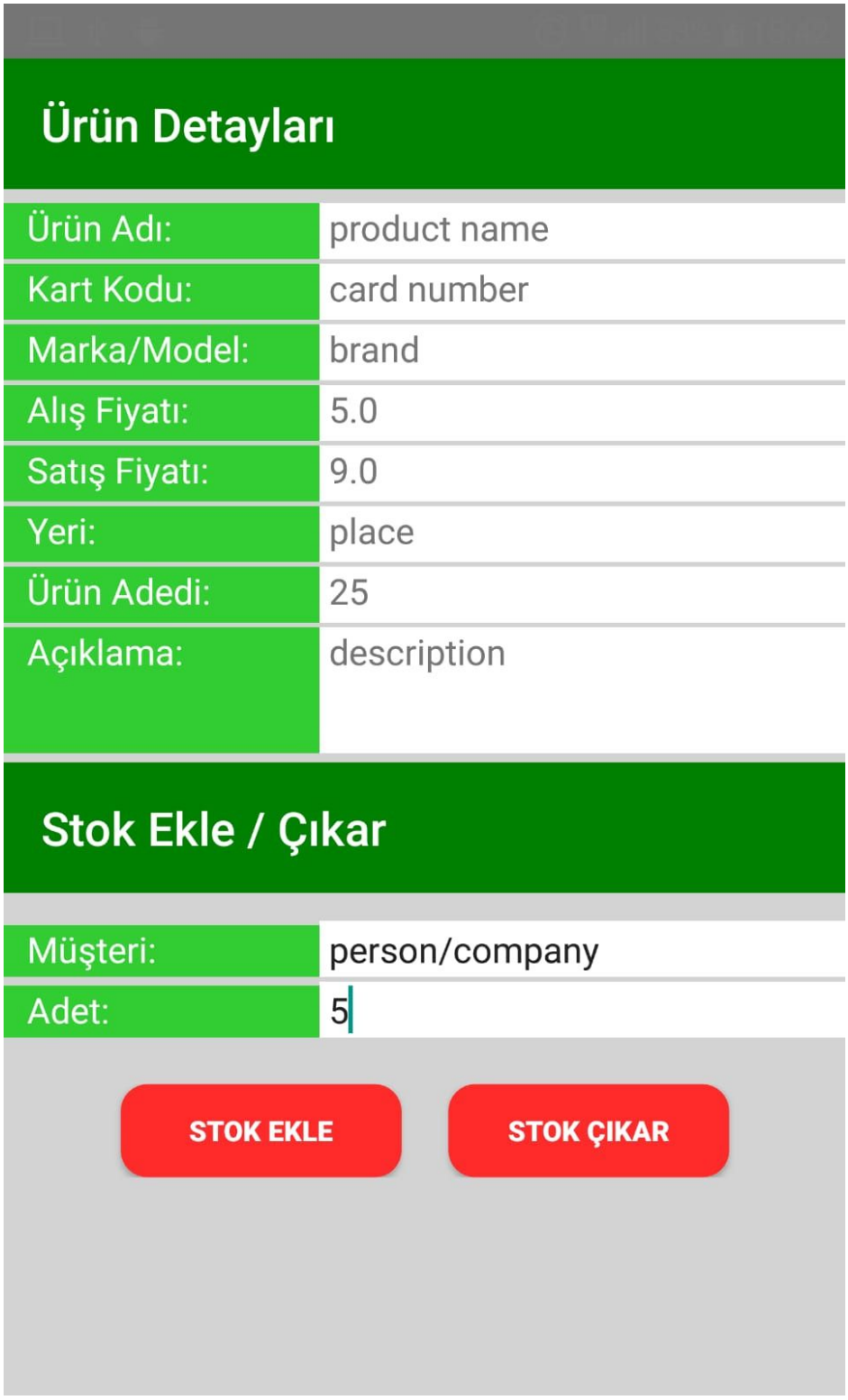
e) Update Product:

User can update any information he/she wants on this page except Card No.



f) Sell / Buy Product:

User will be able to sell & buy product at this page. Application keeps track of number of products and person/company name.



work-sharing between team members:

INTRODUCTION	**Tayfun YURDAER**
DESIGN CONSIDERATIONS	**Tayfun YURDAER**
SYSTEM DESIGN	**Hakan YALÇIN**
DATA DESIGN	**Hakan YALÇIN**
USER INTERFACE DESIGN	**Erkan GÜNGÖR**
CLASS DIAGRAM	**Erkan GÜNGÖR**