Requirements and Analysis Document For Shedbolaget

Samuel Kajava, Emil Svensson, Pouya Shirin, Daniel Rygaard, Daniel Eriksson

Contents

1 Introduction

Shed is a project that aims to extend the functionality of Systembolaget's website with additional features in a desktop application. Most of the work will be in simply copying what Systembolaget already does. We need to focus on minimal browsing first, to catch bugs and be sure that it will work. Then we will reimplement the rest of Systembolaget's features, to lastly add new features. We will add new features to appeal primarily to students, who are poor, have lots of time on their hands and would like to try every kind of alcohol (or at least every kind of beer) once.

1.1 Purpose of the application

The purpose of this project is to recreate a similar version to Systembolaget.se with additional functionality. The user is able to mark products as favourite, sort them with custom sorting methods and product variables. It is designed to extend the functionality of Systembolaget in some aspects.

1.2 Scope of application

Recreating Systembolaget's functionality is not an easy task. In addition to adding new functionality, Shedbolaget will not be recreating the shopping aspect of Systembolaget, where you'll be able to buy the products online, as well as some other smaller features.

1.3 Objectives and success criterias of the project

- List all of Systembolagets products and display its information.
- Save favouritized products locally.
- Display products accordingly to the user's needs.

1.4 Definition, acronyms and abbreviations

- APK (Alkohol per Krona) - Alcohol per Crown

2 Requirements

2.1 Functional requirements

The user should be able to:

- Browse through all of Systembolaget's products
- Filter products by two level of categories.
- Sort products from highest to low and vice versa.
- Favourite products and check their favourites.

2.2 Non-functional requirements

2.2.1 Usability

The application's primary language is Swedish and will be designed to target an adult audience. With a single navigation bar, the application is designed to be simple and reliable.

2.2.2 Reliability

The project fetches its data from the original website - Systembolaget. The accuracy of the displayed data is not 100% since an older version of the data is used.

2.2.3 Performance

The application should be running with as little stuttering as possible.

2.2.4 Supportability

Application should be able to be run on Mac, Windows and Linux platforms.

2.2.5 Implementation

TODO

2.2.6 Packaging and installation

TODO

2.2.7 Legal

TODO

2.3 Application models

High priority use cases:

- Show specific product
- Show all beverages

- Search for a beverage
- Poular beverages

Medium priority use cases:

- Popular beverages
- Detailed beverage information

Low priority use cases:

- APK information
- Drink generator
- Favourites

2.3.1 User case: Show specific products

As a user, I want to select specific beverage types to view so that I can see all beverages within that selection.

Acceptance:

- Can I select a specific type and show all products which share that property?
- Can I select several types at once and view all relevant products?
- Can I go the other way and exclude products?

Tasks:

- Show options for what to select.
- Show what property is selected.
- Show products within the selection.

Priority: High.

2.3.2 User case: Show all beverages

As a user, I want to be able to browse all beverages so that I can get an overview of what's available.

Acceptance:

- Can I view all products in the app?
- Can I see an image of each product?
- Can I see specific information about the product?

Tasks:

- Show all beverages.
- Show the price of each product.
- Show a thumbnail of each beverage.

Priority: High.

2.3.3 User case: Popular beverages

As a user, I want to see new beverages as soon as I launch the application because I often want to find drinks when I'm unsure what to order at the bar.

Acceptance:

- Can I see images of new beverages without much effort?
- Can I easily see how much a new beverage costs?

Tasks:

- Show new beverages.
- Show thumbnail image of the popular beverage.
- Show price of the popular beverage.

Priority: Medium.

2.3.4 User case: Search for a beverage

As a user, I want to be able to search for a product and find information about it so that I can quickly look up specific beverages.

Acceptance:

- Can I search for specific products in the range of products?
- Can I use the search function to filter out products in a specific category?
- Am I able to search for products anywhere in the application?

Tasks:

- Show search box.
- Allow the user to enter a search term in the search box.
- Show results based on the search term.

Priority: Medium.

2.3.5 User case: Detailed beverage information

As a user, I want to see detailed information about a product, so that I can learn more about products that interest me.

Acceptance:

- Can I view an image of the product?
- Can I see:
    - How much it costs?
    - Product APK?

– Product volume?

– Alcoholic percentage?

– Product description?

- Can I obtain this information from anywhere where there are products?

Tasks:

- Show search box.

- Allow the user to enter a search term in the search box.

- Show results based on the search term.

Priority: Medium.

2.3.6 User case: APK information

As a user, I want to be able to find the products with the highest APK in a leaderboard so that I can see which products are the most cost efficient in terms of alcohol contents.

Acceptance:

- Can I view the products with the highest APK?

- Can I choose which product types to view when I'm looking at APK values?

- Can I access an item and get more information about it from the list?

Tasks:

- Show the leaderboard.

- Show the products with the highest APK ratio.

- Allow the user to choose which type of beverages are shown.

- Show which type of beverage each product is as well as its name, alcohol contents, volume, apk and price.

Priority: Low.

2.3.7 User case: Drink generator

As a user, I want to be able to find drinks based on products I select so that I can find new drink recipes to try out.

Acceptance:

- Can I choose some ingredients and get new drinks with added ingredients?

- If I'm not satisfied with my result, can I retry?

Tasks:

- Allow users to select some known ingredients.

- Generate new drinks based on the known ingredients.

- Show the generated drinks on screen.

Priority: Low.

2.3.8   User case: Favourites

As a customer, I want to be able to mark some products as favourites because I want to remember to keep buying those.

Acceptance:

- Can I mark a beverage as a favourite?
- If I change my mind, can I unmark favourited beverages?

Tasks:

- Allow user to mark beverages as favourite.
- Allow the user to view all marked beverages.
- Allow the user to unmark beverages in case they change their mind.

Priority: Low.

2.4   Definition of Done

- All acceptance criteria are met
- All public methods has unit tests.
- All Travis checks pass.
- Completed Java doc (if applicable)
    - All new files need to have an author.
    - All public methods except getters and setters has Javadoc.
- Pull request is accepted by everyone in the group

2.5 User interfaces
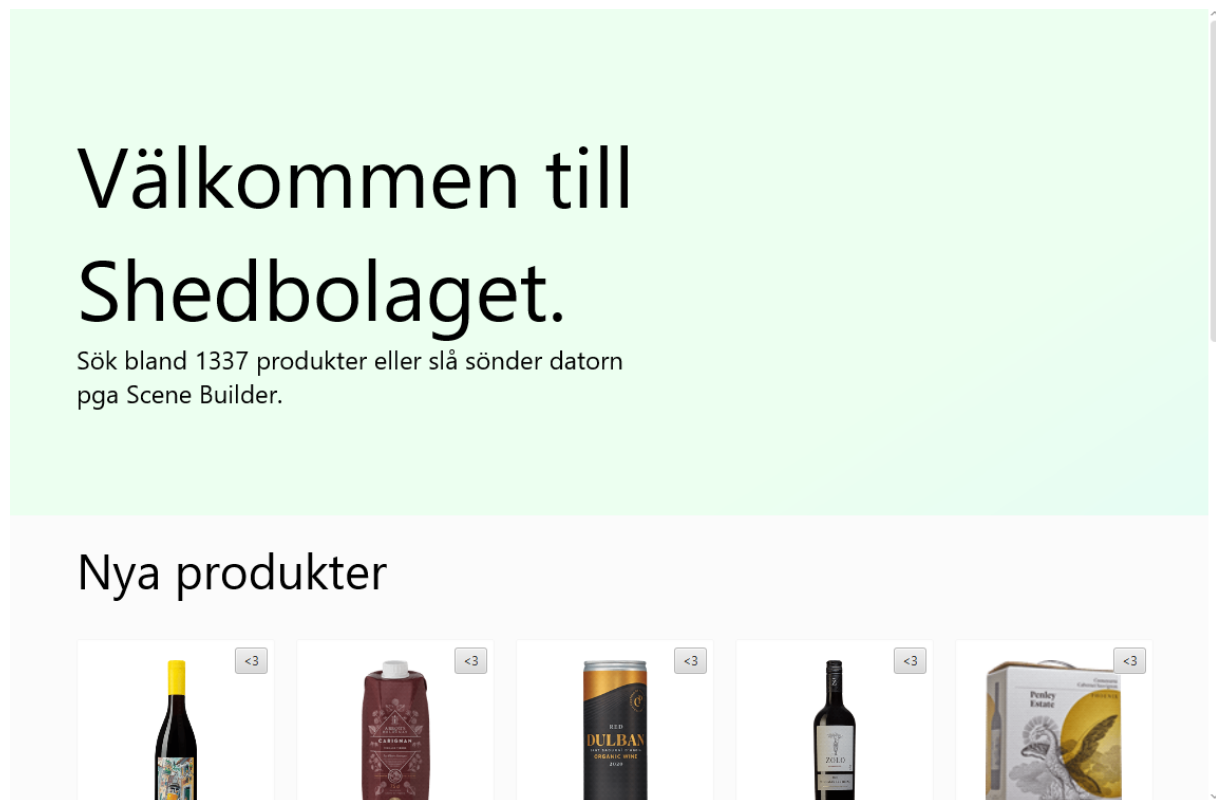
Figure 1: This is the starting page for the application. A welcoming message with products displayed at the bottom part of the application.
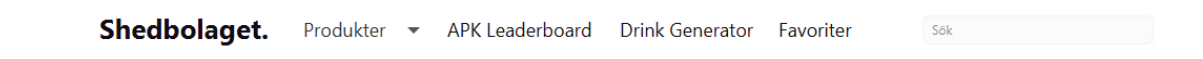
Figure 2: This navigation bar is shown at the top of the application. This bar is used for almost all of the navigation paths. The Escape Hatch included in the top left corner and the Search bar in the opposite corner. Between these, the user will find the necessary links to the main pages of the application.

Figure 3: Further down in the start page, is a grid of products under the label "New Products".

Figure 4: Entering the products page, will be a list of products on the right, and a choice of secondary categories to the left.
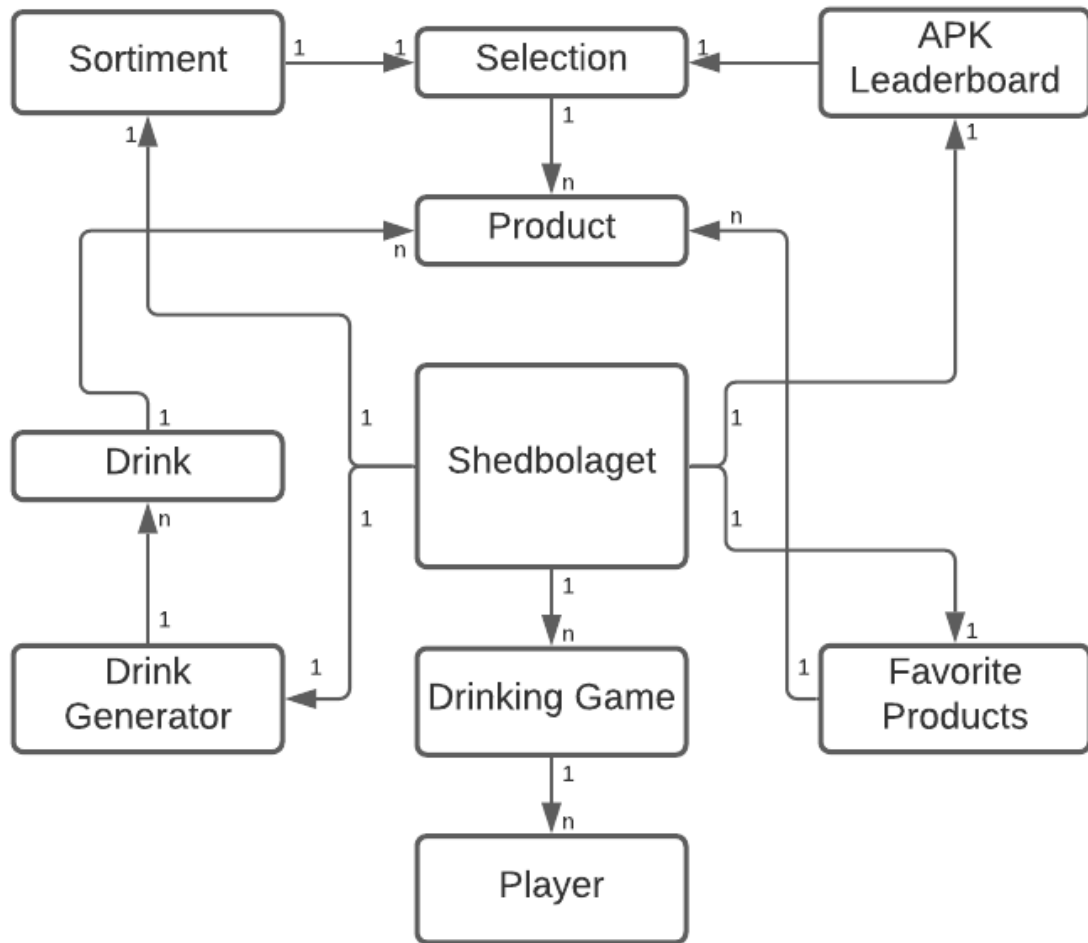
3 Domain Model

Figure 5: The Domain Model

3.1 Class Responsibilities

As observed in the domain model image, all class modules work independently with the exception of the references to the Product class. All modules are bound to the product class in some way*, since the whole program is built upon the product data. Each of the modules have their own area of responsibility where they together complete a whole program. The model class 'Shedbolaget' is the main class that brings the modules together to create this desired program.

*Drinking games module not included, since it's a feature not yet to be implemented.

4 References