Stomanage

Software Design Document

Prepared by

Stav Nof

Yarin Shapira

Date: 15/01/2021

TABLE OF CONTENTS

1. INTRODUCTION 3

1.1 Purpose 3

1.2 Scope 3

1.3 Overview 3

2. SYSTEM OVERVIEW 4

3. SYSTEM ARCHITECTURE 4

3.1 Architectural Design 4

3.2 Decomposition Description 5

3.3 Design Rationale 5

4. DATA DESIGN 5

4.1 Data Description 5

4.2 Data Dictionary 6

5. COMPONENT DESIGN 6

6. HUMAN INTERFACE DESIGN 7

6.1 Overview of User Interface 7

6.2 Screen Images 7

6.3 Screen Objects and Actions 8

7. REQUIREMENTS MATRIX 8-9

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of solving the problem of managing the warehouse inventory and the precise implementation and details required to satisfy the requirements as specified in the Software Requirements Specifications (SRS). It is assumed that the reader read the SRS, since this document also defines the implementation details of the design behavior given the requirements within it.

1.2 Scope

Background - Within the warehouse staff, there is a division into roles and a hierarchy of responsibilities. For example, some of the staff members are responsible to make an inventory count and making sure that all the equipment signed is returned at the end of each day. Also, some are responsible for performing an annual inventory count, some are responsible for ordering new and missing equipment (dealing with the scout groups money), and so on.
The above division makes it possible to maintain order and organization and moreover (and perhaps even more importantly) and provides meaning and a way to advance in the hierarchy to the manpower of the warehouse staff.

A desirable feature of our system is the ability to classify certain actions in the system so that not every team member can perform it unless he is in a role that provides him access to the specific action
A reliable and synchronized – the system will show the status of all the equipment that was fed to it and will sync in real time, so that if a number of guides order a specific item and it runs out while the order is being taken the guide will get a real time notification.
The system will produce an updated image to the request of the manager based on selected criteria.

1.3 Overview

The design description defined in this document serves multiple purposes:

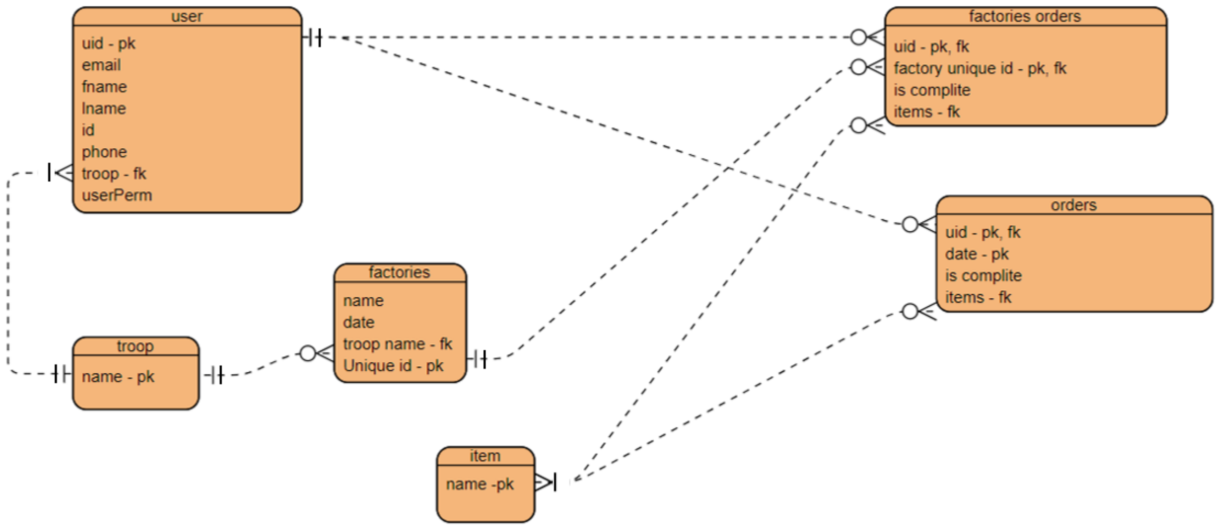
- To describe the functional structure, data and algorithms to be implemented.
- To identify required system resources.
- To be used to assess the impact of requirement changes.
- To be used to verify compliance with requirements.
- To aid in maintenance activity.

2. System Overview

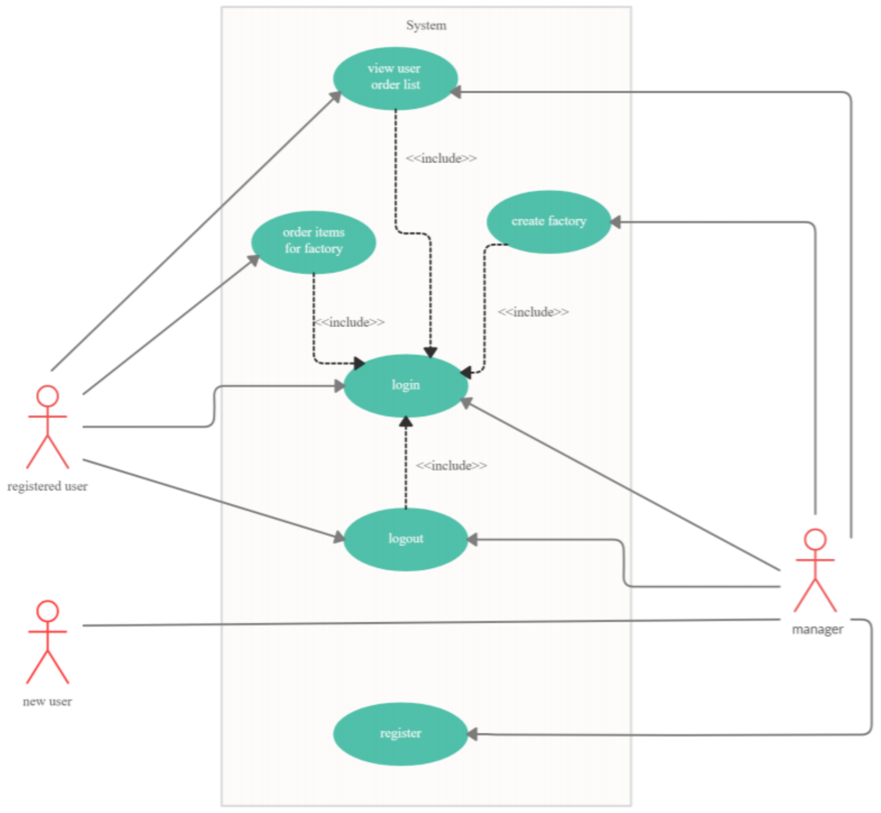
Our inventory management system will allow the organization of a complete and broad picture of the inventory when counting inventory, will enable ordering and signing equipment with the help of a user-friendly website/app that allows full transparency in real-time for all users.
Also, the system alerts and signals to the staff members when inventory is about to run out and produces outputs that will help the organization become more efficient, such as what equipment runs out the fastest, which is ordered the most, or is destroyed the fastest, etc.
The system will have two different types of users with different authorizations. The main user – the staff team. This user type will have physical access to the inventory that is inside the storage and therefore will be able to edit, add or subtract from the database's inventory levels at any time, in addition this user type will be able to enable the option to make orders that will be made by the guiding teams. Another important ability that this user type will have is to manage the users of the guiding teams, as in add or subtract users from the system. The main user will be able to view every order that was ever made, even expired orders, and edit them.
Another ability is to export data and information about the inventory status.
The sub user is the guiding team's user. This is authorized to order equipment from the storage for every non – expired activity that was opened by one of the managers.
Additionally, this user will be capable to access all of the orders he made, view and edit his order history (even expired).

3. System Architecture

3.1 Architectural Design



3.2 Decomposition Description



3.3 Design Rationale

The rationale for selection the architecture described in 3.1 section was to create an independent component for each subsystem this way the components will have a separate functionally and will reduce the resources cost. Using this architecture will improve the system modularity and makes it more flexible for changes as opposed to using a single component.

4. Data Design

4.1 Data Description

The information will be transformed into class object entities and will be stored in a database. For the analysis and efficient search the database that will store the users with premissions, items list, orders for activities, orders for factories and groups list.

4.2 Data Dictionary

Entities list:

- Users – this list shows all the users that have been integrated into the system
- Inventory – this list accounts for the inventory in the warehouse
- Activity orders – this shows the orders made by the users.
- Factory orders – this list shows all the factories the admin listed for orders
- Tribes – this list shows all the tribes that were entered into the system

5. Component Design

Admin – warehouse staff

- New user
  - Register new users
  - Grant permissions
  - manage current users – change permissions
  - Delete users
- Tribe managment
  - Add tribes
  - Assign factory to scout group
  - Delete tribes
- Inventory managment
  - Add items
  - Edit items
  - Delete items
  - Uploat images of items (for convinience)
- Factory and activity managament
  - Open new factory based on tribe
  - Watch fasctory orders
  - Watch activity orders
  - Delete factory

User – guide

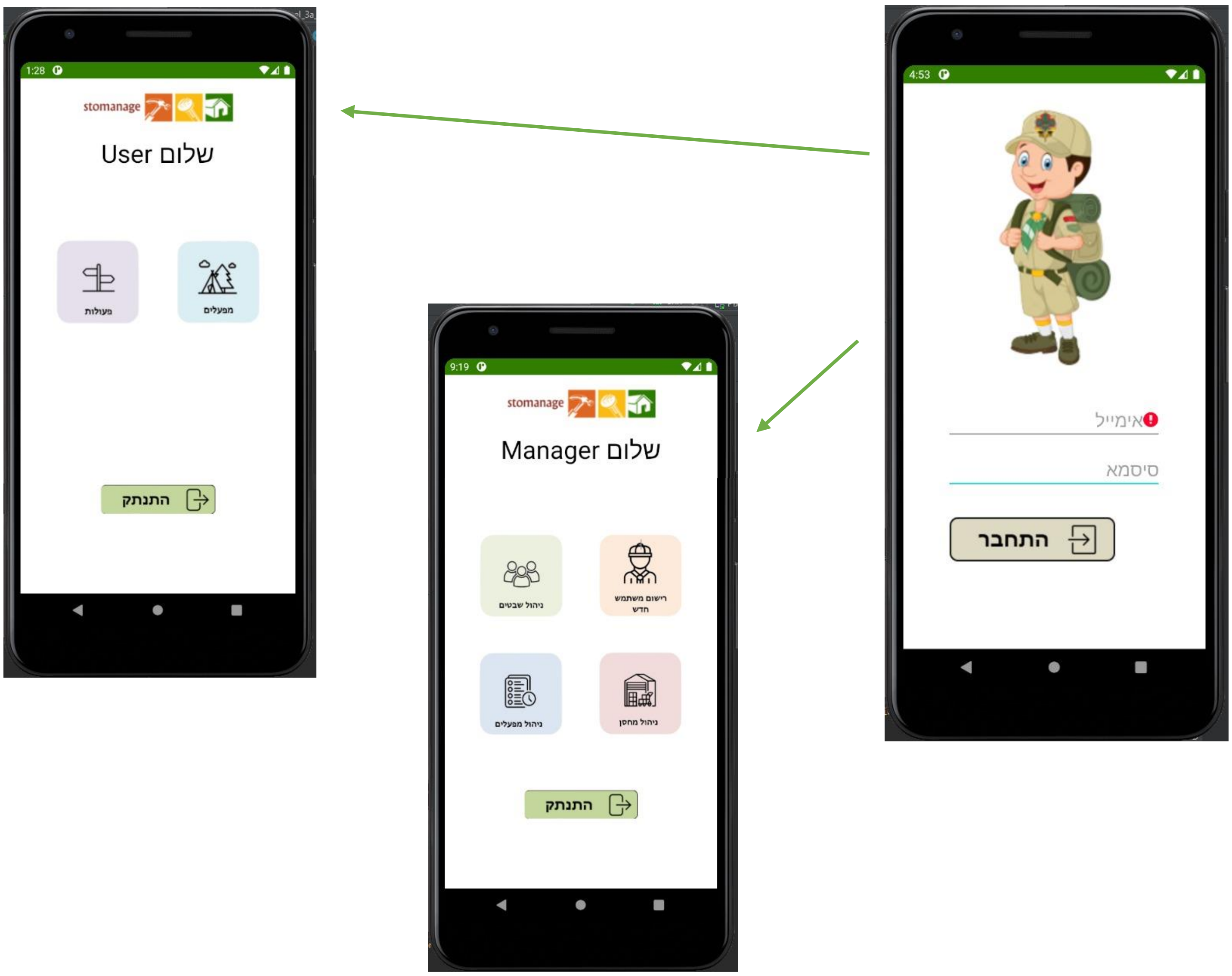
- Actions
  - Watch the activity archive by date
  - Edit orders
  - Delete orders
  - Open new order
- **Factories**
  - watch the factory archive (both open and closed). The list of factories will change according to tribe
  - edit open orders (status changed by the admin)

6. Human Interface Design

6.1 Overview of User Interface

- Home screen – admin
  - Register new user
  - Manage tribes
  - Manage warehouse
  - Manage factories
- Home screen – user
  - Factories
  - Activities

6.2 Screen Images



6.3 Screen Objects and Actions

Everyone

- Login page

The app has two types of permissions

- Admin – open and manage factories ,register new users ,watch and change status of orders
- User – order and edit items for activities and factories ,watch the order archive

After the login page ,the user is directed the homepage suited for his user type (user or admin).

The functions that appear in each home screen type are listed in section 5 of this document.

7. Requirements Matrix

components	Subtype	Non\functional	Permission	Requirements	
all	Performance	Nonfunctional	Software	Each page loads in less than one second	1
Manage factories And activities	data	functional	user	Any order can be saved / deleted even if it has not been completed and if it has not been closed by the manager	2
Activities factories	data	functional	user	Any order can be edited / modified if it is not closed by the manager	3
Activities factories	data	functional	user	Orders from previous unsubscribed activities can be accessed	4
users	Security	Nonfunctional	user	Each user has a username and password	5
all	Performance	Nonfunctional	Software	Any user can access from any Android device	6
all	Performance	functional	user	Any member of the Scout movement can connect / disconnect from the app	7
Manage items list	Performance	functional	user	Any user can remove / add equipment from stock to order	8
Manage factories	Performance	Nonfunctional	Software	Each activity has a pre-set date range for ordering equipment	9
Activities factories	Performance	Nonfunctional	Software	Each order has a system entry date	10
Activities factories	Performance	Nonfunctional	Software	Each order has a last update date	11
users	data	functional	user	Each user has their own characteristics such as scouts group name and mail	12
all	Performance	Nonfunctional	Software	The app will use a list data structure	13

all	Safety	Nonfunctional	Software	The app will run smoothly	14
all	Safety	Nonfunctional	Software	The app will run 7/24	15
Activities factories	Performance	Nonfunctional	User	Each order can be shared with other users	16
Activities factories	data	functional	User	Each order can be named	17
all	requirement	Nonfunctional	Software	The app will run without a budget	18
all	requirement	Nonfunctional	Software	The app will require the use of the Android platform	19
Manage items list	Performance	functional	User	Anyone in the warehouse staff can add / delete equipment from stock	20
reports	Performance	functional	Software	The app will generate reports on existing / missing inventory quantities	21
all	Safety	Nonfunctional	Software	The application will automatically synchronize with the database	22
all	Safety	Nonfunctional	Software	The system will automatically save the status of the user	23