SRS - Software Requirements Specification

**Table of Contents**

1. Introduction………………………………………………………………………………….

1.1. Purpose…………………………………………………………………………

1.2. Scope…………………………………………………………………………….

1.3. Definitions, acronyms, and abbreviations……………………..

2. Overall description……………………………………………………………………….

2.1. Product perspective & functions…………………………………..

2.2. User characteristics………………………………………………………

3. Specific requirements………………………………………..…………………….…..

3.1.1. User interfaces – screenshots……………………………..

3.1.2. Hardware interfaces…………………………………………..

3.1.3. Software interfaces…………………………………………….

3.1.4. Communications interfaces…………………………..........

3.2. Functional requirements – Use Cases …………………………………………..

3.2.1. General………………………………………………………………

3.2.2. Operator……………………………………………………………..

3.2.3. Administrator……………………………………………………..

3.2.4. Hardware……………………………………………………………

3.3. Performance requirements…………………………………………….

3.4. Design constrains……………………………………………………………

3.5. Software system attributes…………………………………………….

1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1. Purpose

The purpose of this document is to give a detailed description of the requirements for the “Smart Shooting Range” (SSR) software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system.

1.2. Scope

SSR is a system that integrates and connects three main end units: Central Server, Micro controller, Management application. Each end element will be programmed in a different language and environment. Shooting-Range operators can provide their information using the management application. This information will act as the bases for the search results displayed to the Shooting range club owners. An administrator uses the mobile application in order to administer the system and keep the information accurate. The administrator can, for instance, verify costumers and manage user information. Furthermore, the software needs both Internet and WIFI connection to fetch and display results. All system information is maintained in a database, which is located on a web-server. The software also interacts with the Arduino software which is required to be an already installed software on the MC in the "filed". By using the MC, operators can view hardware feedbacks in real time on the app map.

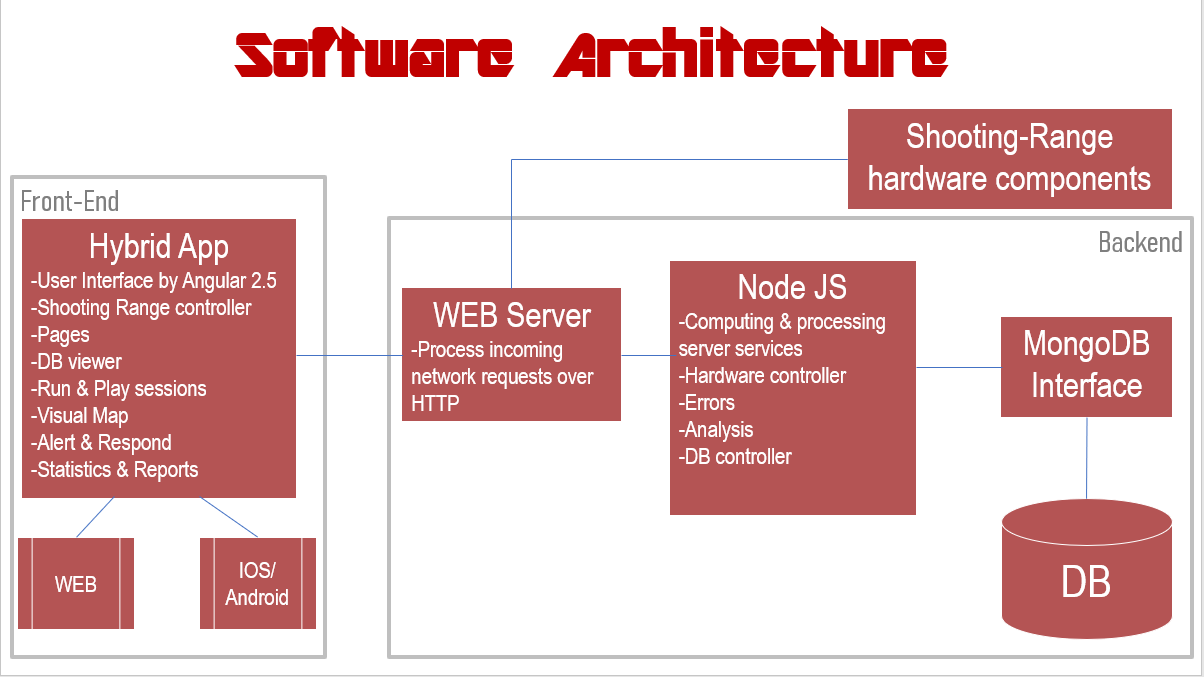
1.3. Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| SSR | Smart Shooting Rang |
| SR | Shooting Range |
| MC | Micro-Controller |
| Warrior | Someone who using the SR as a shooter, can be a costumer or guide. |
| Operator | Someone who run the SR scenario, using the management app. |
| Administrator | System administrator who is given specific permission for managing and controlling the system. |
| Web-Server | A server which provide RESTful API. |
| RESTful API | an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data. |
| Node-JS | Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. |
| Node-JS server | handling lots of events asynchronously. |
| ESP | The chosen micro-controller having WIFI module. |
| Docker | Docker is a computer program that performs operating-system-level virtualization, also known as "containerization" |
| Docker Container | relies on the kernel's functionality and uses resource isolation for CPU and memory and separate namespaces to isolate the application's view of the operating system |
| Arduino | Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers |
| Front-end | The mobile application. The part of the system that the operator can see and interact with. |
| Back-end | The server services – web-server, database, and other processing services. |

2. Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1. Product perspective & functions

This system will consist of three parts: one front-end (mobile) application, back-end server and MC code.

**The mobile** **application** will be used to: observe SR scenarios, reports creation, information about warriors and more. The mobile application will communicate with backend services using web-server. Also, the mobile app will allow the operator to control the SR hardware by giving them a prior instructions, for instance, "sensor #1 activate target #3". Mobile app will provide a graphic map, a top view that show the SR-arena current status, videlicet all the SR components positions.

**The backend services** will supply database service, REST-API, data processing, data supplier, time reference service and will function as an integrator between the micro-controller and the SR hardware components to the mobile app. With these services the SR operator be able to instruct the MC with bunch hardware instructions about the SR scenarios, and issue desirable reports, for instance, statistics, warrior data and follow-up training process of a warrior.

**The micro-controller code** will allow a real-time control of the SR components. The MC will act as a base station that links all the SR components and will manage their work. The MC decides when and which to activate. During the session, the MC gathers and organizes all the information and eventually send it back to the backend server.

2.2. User characteristics

There are two types of users that interact with the system: users of the mobile application - operators and administrators. Each of these has different use of the system so each of them has their own requirements.  
The mobile application operators can use the application to manage SR scenarios. This means that the operators have to be able to search for warriors information, choose a and arrange SR map and then send away the MC instructuons, start and stop scenarios, watch times and shown wanted reports.   
The Administrators will not use the mobile application but the backend services instead. There they will manage the information about their SR, warriors and scenarios, for example a description of the warriors, training processes and etc. The administrator can manage the information for each warrior or team.

**3. Specific requirements**This section contains all the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

**3.1. External interface Requirements**This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

**3.1.1. User interfaces**

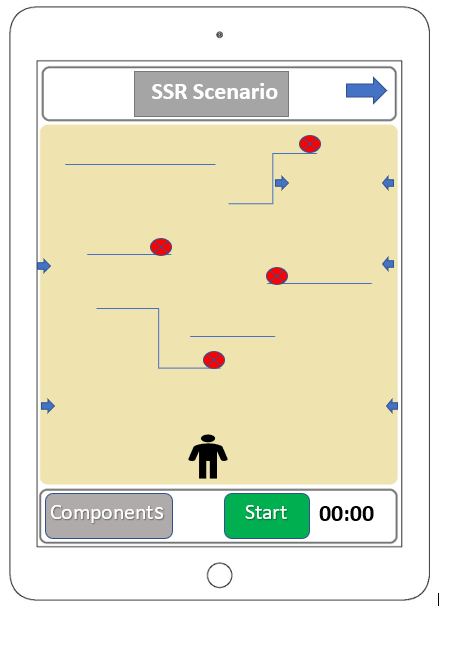
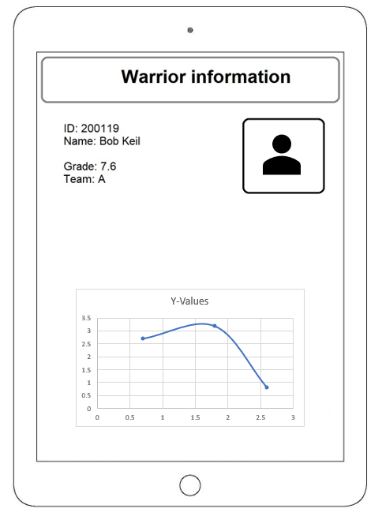
First, user of the mobile application (operator) should see the home project page, see Figure 1 when he/she opens the application the app set communication with the backend server and check for "healthiness". If the connection fails, an error message will appear. From the home page the operator can choose one of the displayed. Warrior will open warriors age with an option to search wanted warrior, see Figure 3, add new warrior and edit an exist warrior profile. Every warrior has its own personal information about his training process.

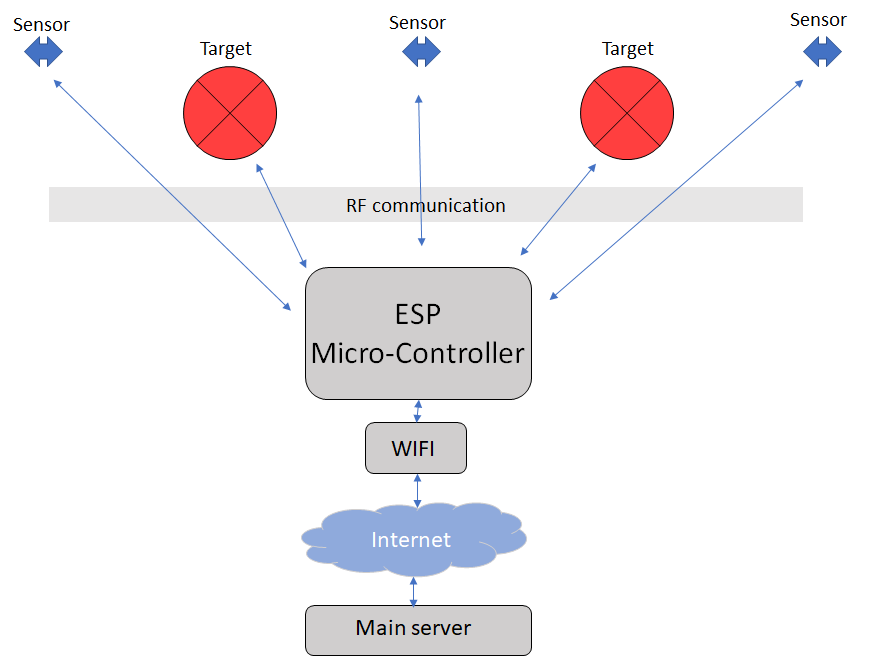
Figure 3

Figure 2

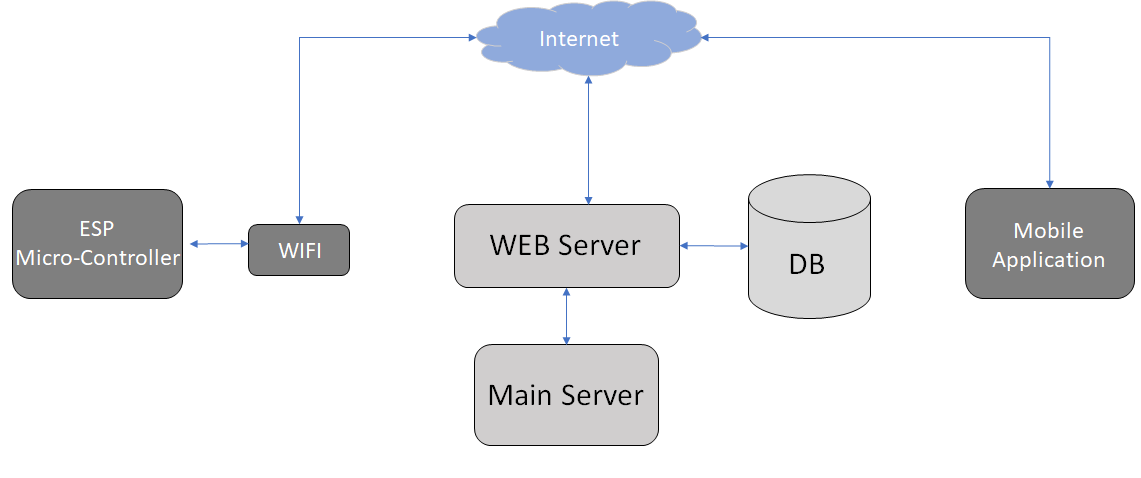
Figure 1

In the scenario page, the operator determine the map status, he set each SR components (targets, sensors, walls) at the current position, in addition he set instruction for each component, for instance, "target #1 activate by sensor #1". When everything ready the operator press "Start". When his press Start, all of the map information (instructions) are send to the backend service that deliver it to the MC in the field.

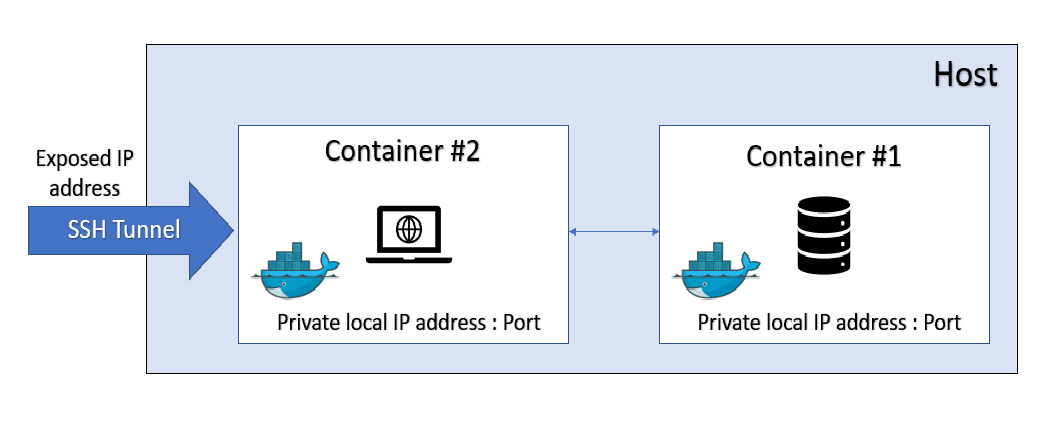
**3.1.2. Hardware interfaces**

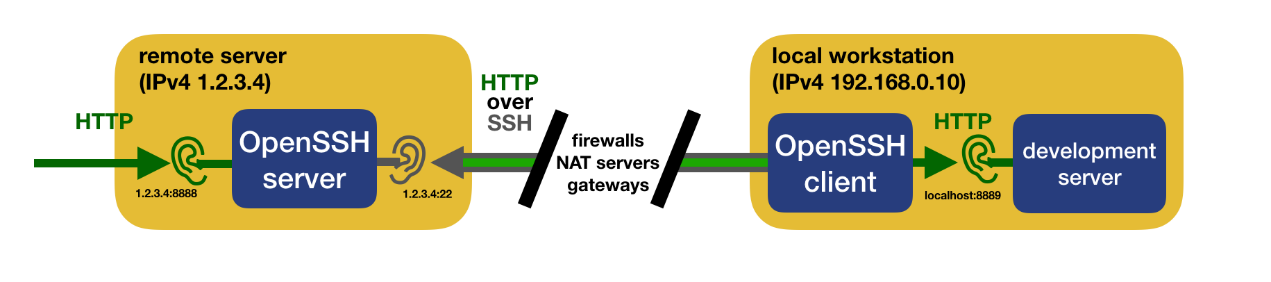
Micro Controller and hardware components should communicate autonomously right after the instructions set sent from the mobile app through the backend server. MC code should be able to polling each sensor and target in the arena, response accordingly and document everything that goes on.

**3.1.3. Software interfaces**

****The user application and the MC use the web server service to communicate, send and receive data. The web server has REST API that provide that functionalities. In addition the web server maintain data, using database, in order to get MC and the app information about any of their relevant data. The communication between the database and the web server consists of operation concerning both reading and modifying the data.

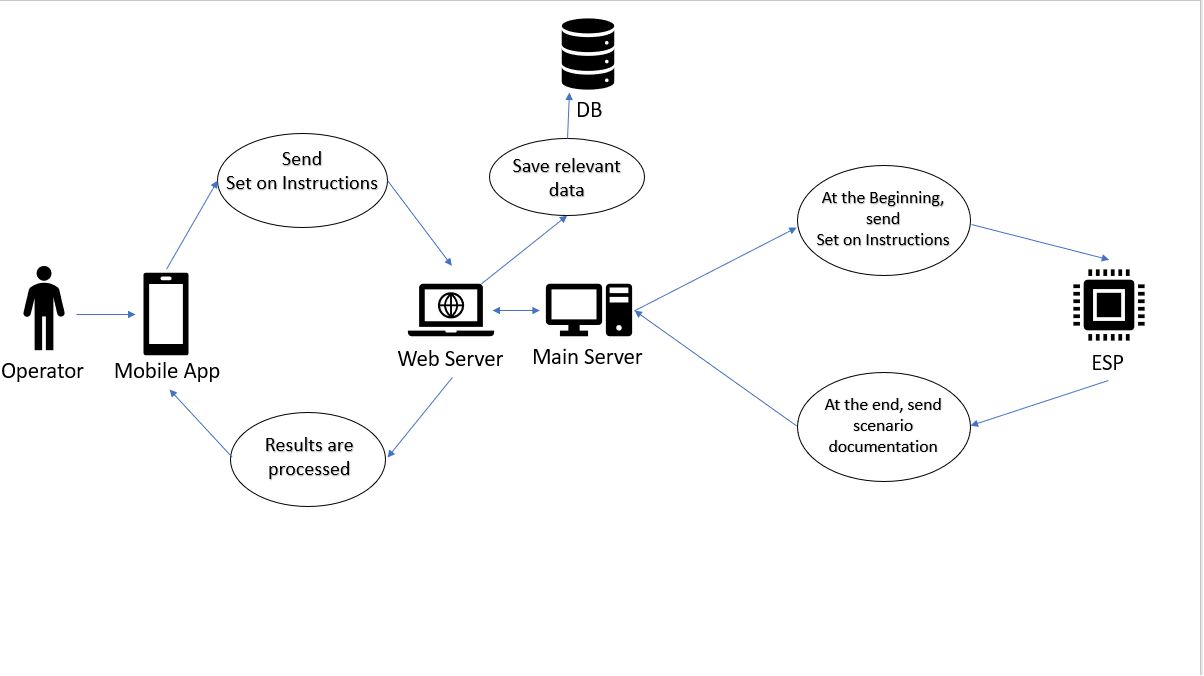
**3.1.4. Communications interfaces**The communication between the different parts of the system is important since they depend on each other. The main communication leaning on a TCP/IP protocol through the global internet network, since the main services are not really physically in the field, they can deploy to any desire machines. In addition, the MC and the mobile application using WIFI chip to connect the internet. The web server (REST-API service), main server and database service are runs on Docker's containers in some host OS machine, and communicate each other in a local bridged network.



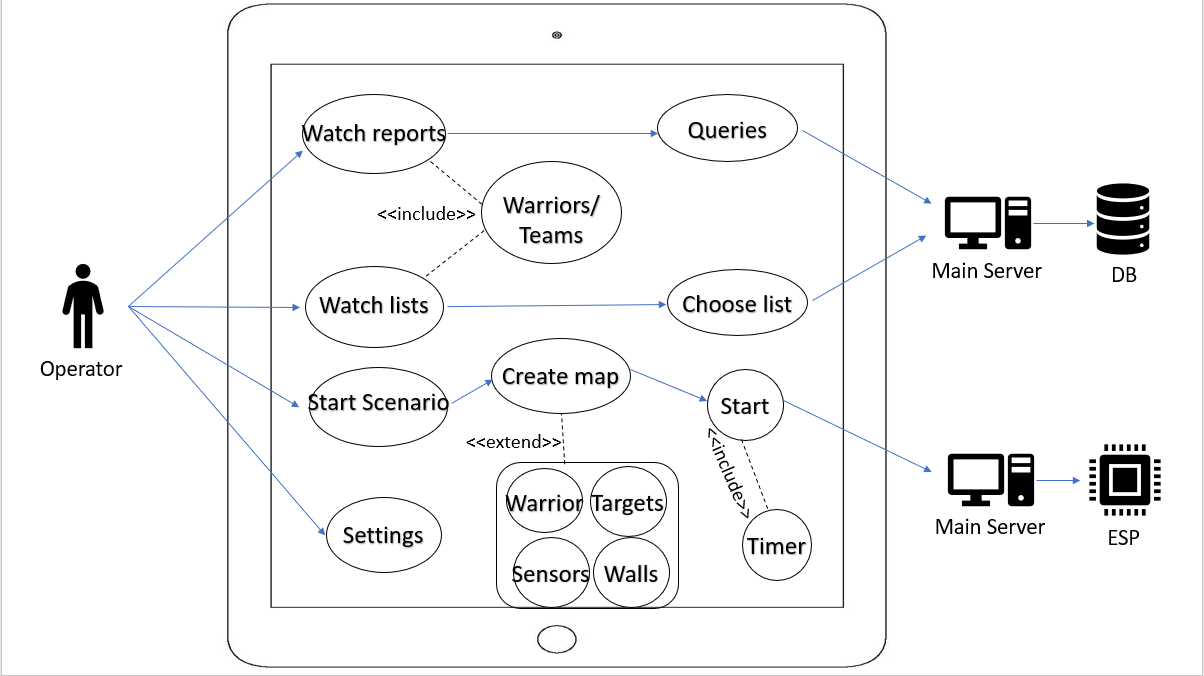
 Important to mention that the web service expose it's IP address through SSH tunneling protocol. The router that stands between our workstation and the internet makes it harder to expose a local socket to the outside. Most of the time, this is preferable for security. A typical home network is composed of a router and a number of devices connected to the router. The router is assigned a public IP address by the internet provider. Every device in the home network, including the router itself, is assigned a private IP address, usually in the form 192.168.x.x. The router acts as a gateway between the home devices and the internet. This gateway performs network address translation (NAT) between the public and the private networks. While this allows a local device to reach a public IP address, it is usually not possible to initiate a communication from the outside.**Tunneling services** Some services offer you to set up a tunnel between their servers and your local workstation. These services can be very easy and convenient to use and provide advanced features such as monitoring and statistics, domain name and HTTPS support. However, these services are either charged or subject to limits and you may want to avoid a third-party accessing your data.

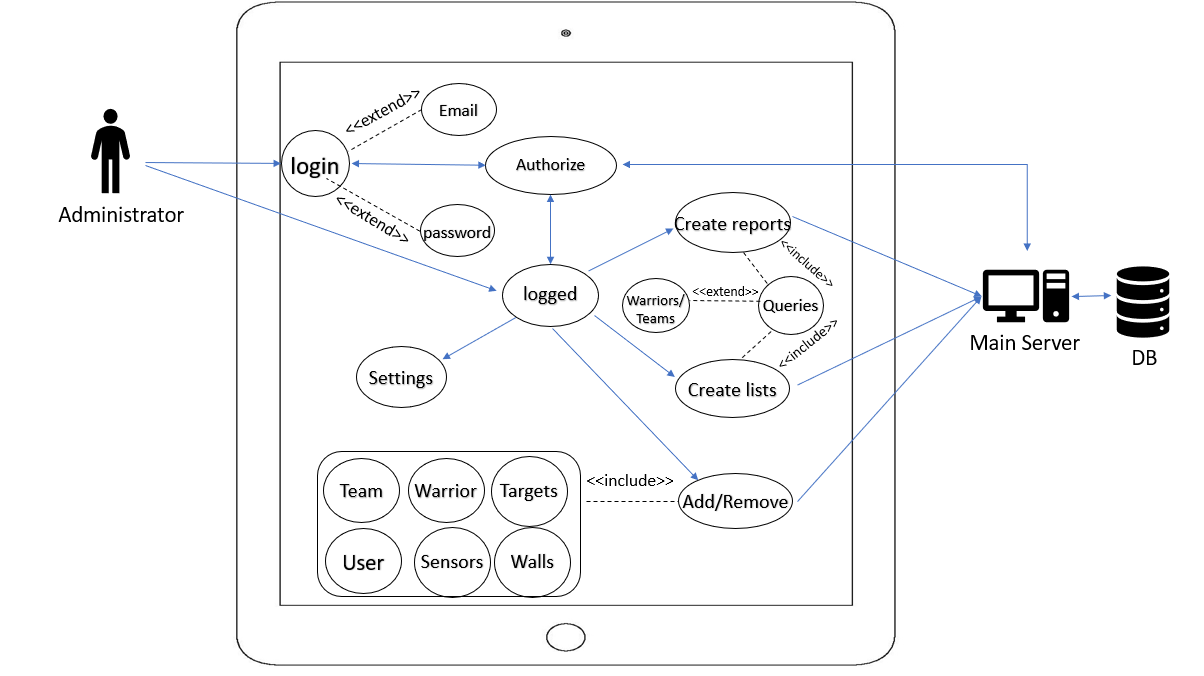
3.2. Functional requirements

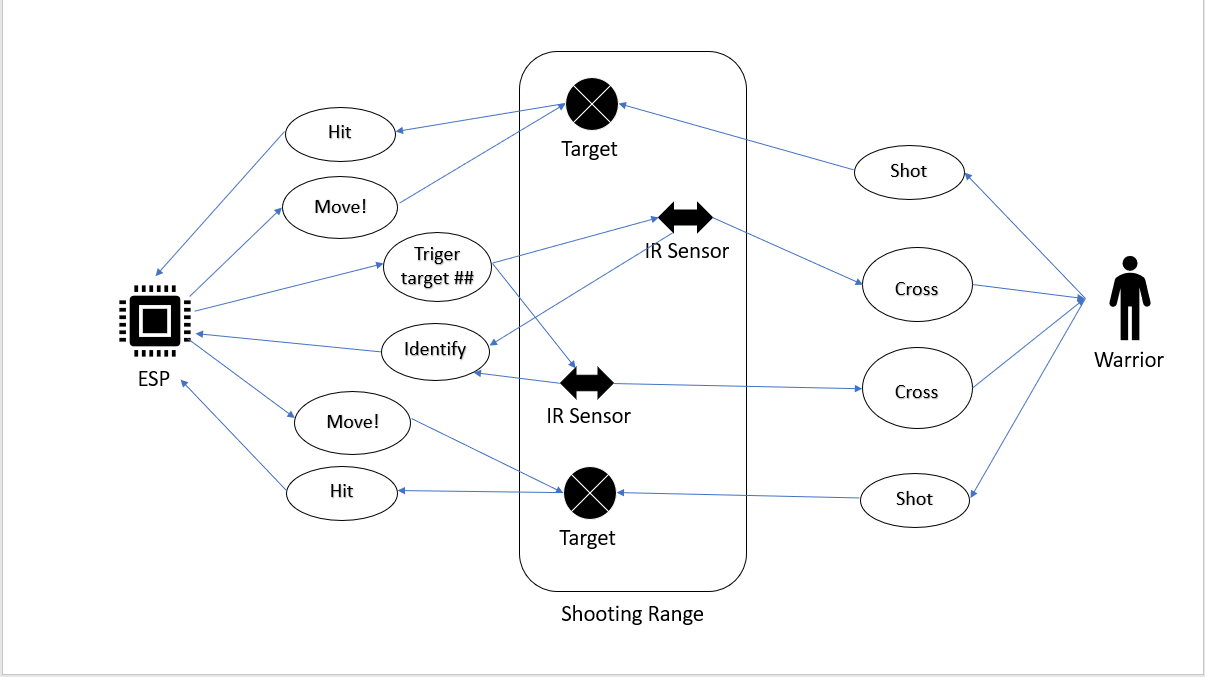
This section includes the requirements that specify all the fundamental actions of the software system describing by **use-cases.**

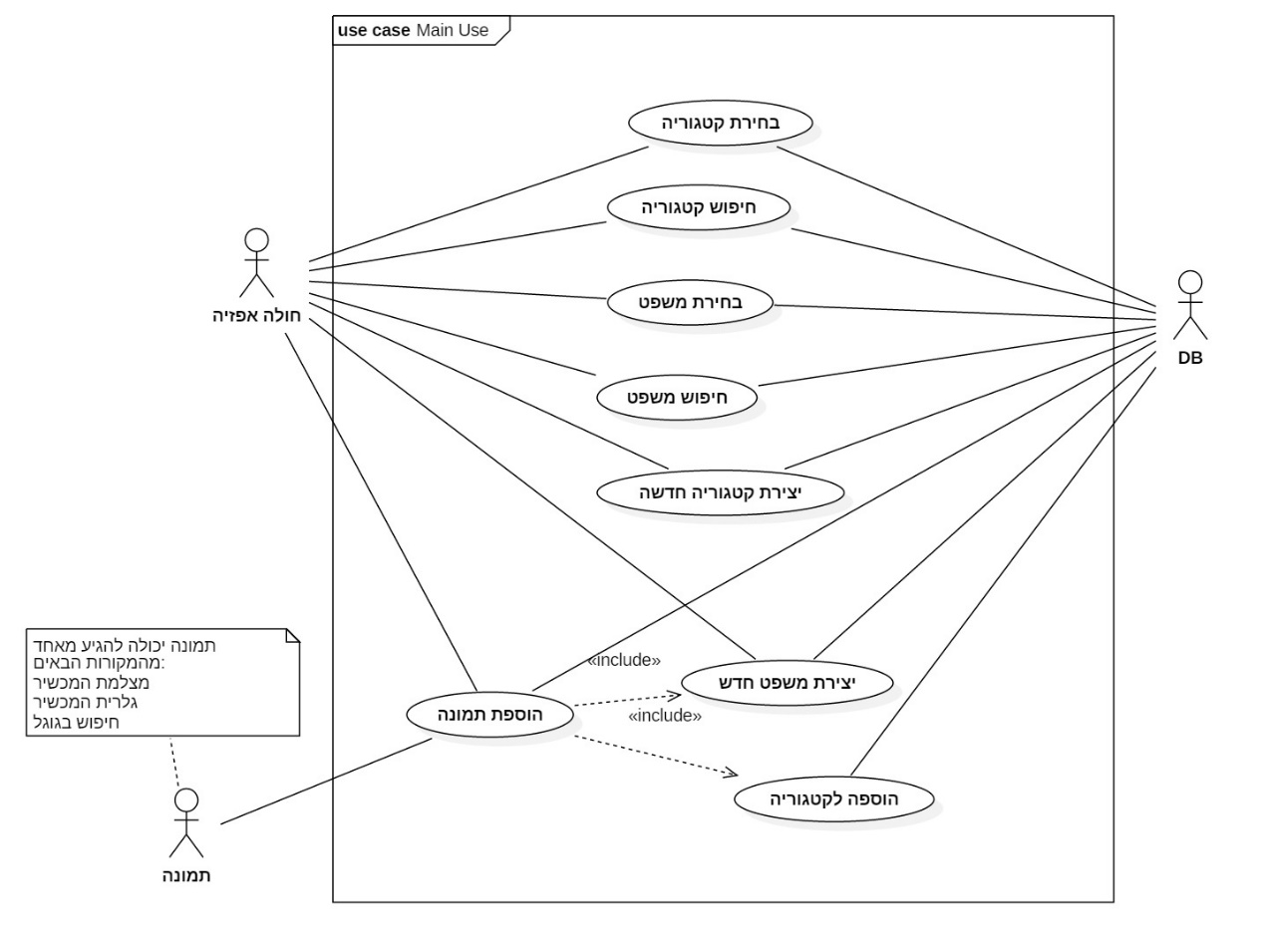
**3.2.1. The General System Use-Case**

**3.2.2. - The Operator Use-Case**

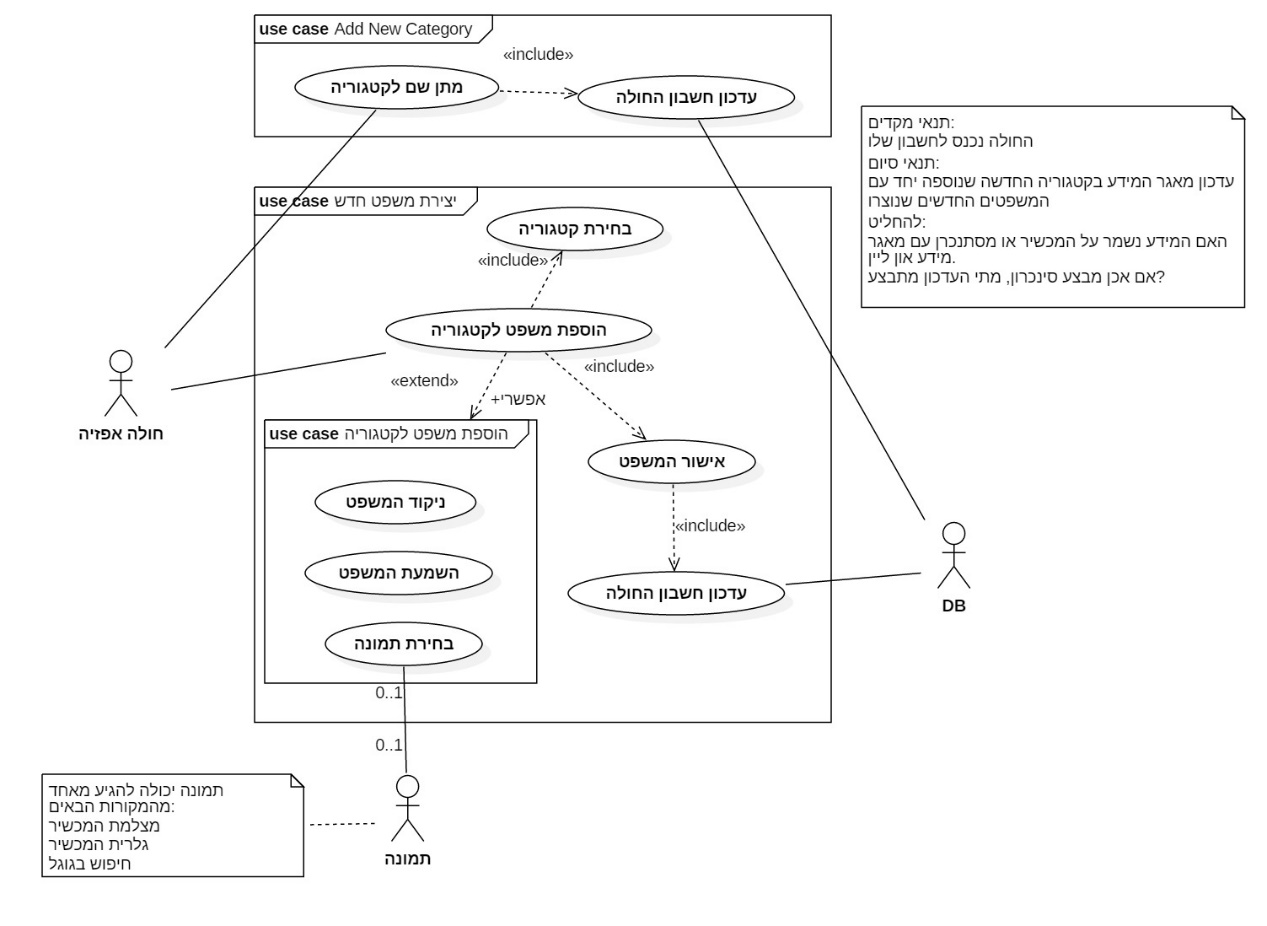


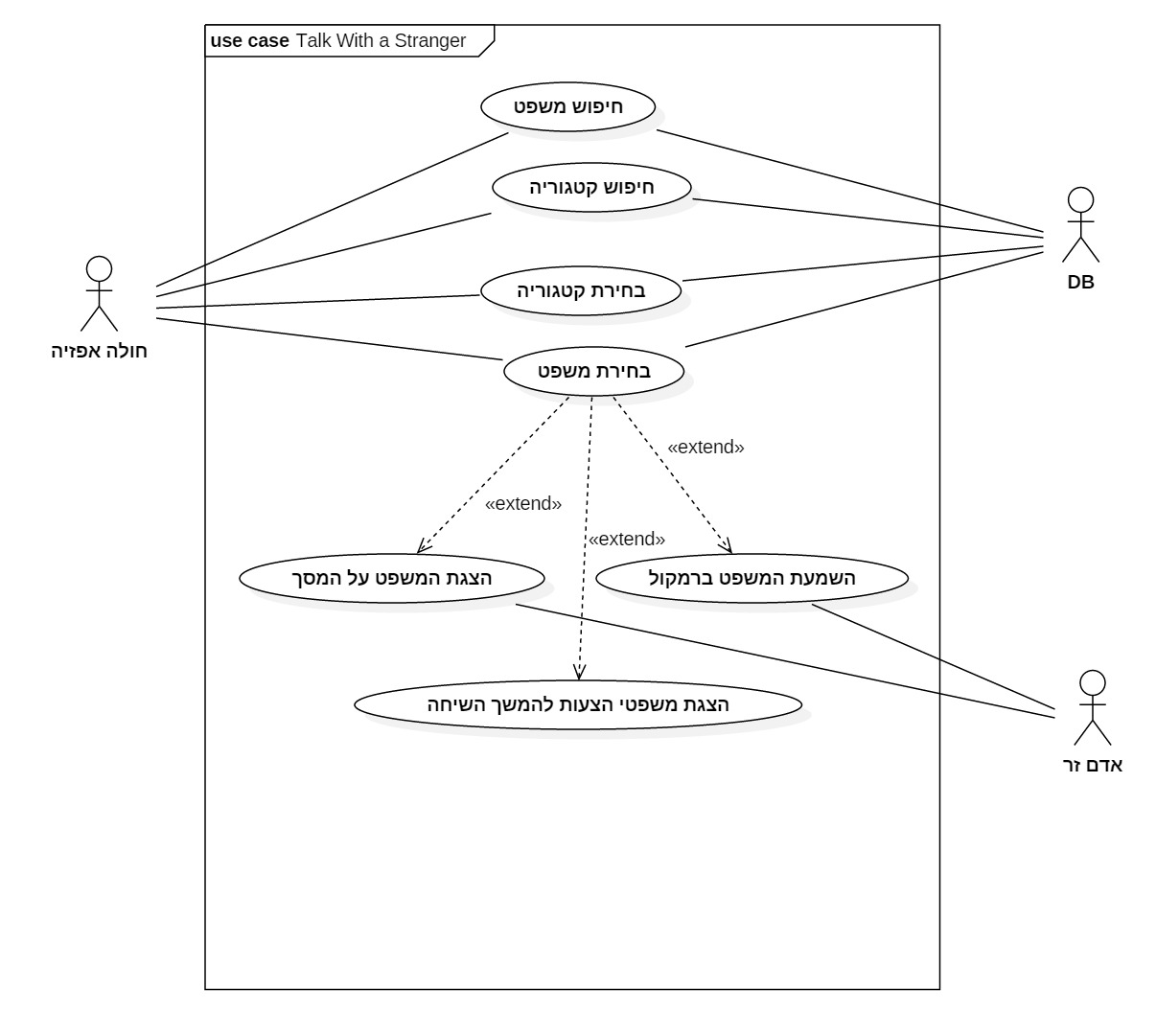
**3.2.3. The Administrator Use-Case**

**3.2.4. Hardware Use-Case**



* 1. **הוספת פריט חדש:**



* 1. **שיחה עם אדם זר:**

1. תרחישי שימוש פורמאליים עיקריים

|  |  |
| --- | --- |
| שם התרחיש | *UC1. תקשורת עם אדם זר – שיחה יום יומית* |
| שחקו ראשי | *חולה אפזיה* |
| מטרה | *ביצוע שיחה עם אדם זר בנושא כללי* |
| היקף ורמה[[1]](#footnote-1) | *מערכת האפליקציה, המשתמש* |
| בעלי עניין ואינטרסים | *המשתמש (חולה האפזיה), ואדם זר הנקלע לשיחה עם החולה.* |
| תיאור | *חולה פוגש אדם רנדומלי ברחוב וברצונו לקשור איתו שיחה בנושא יום יומי, לדוגמא מזג האוויר או הכוונה כיצד להגיע למקום מסויים. ראשית כל, יציג לו את עצמו ואת המוגבלות שלו, ולאחר מכן יפתח עימו בשיחה לפי רצונו.* |
| טריגר | *החולה נכנס למערכת באמצעות המכשיר הסלולרי* |
| תנאי קדם | *החולה עבר את מסך ההזדהות (login)* |
| תנאי סיום מוצלח | *החולה הצליח לתקשר בצורה טובה ומהירה עם האדם הזר* |
| תנאי סיום כישלון | *האדם הזר המשיך בדרכו בגלל סרבול המערכת* |
| **תרחיש הצלחה עיקרי** | 1. *המערכת מציגה מסך ראשי עם תפריטי שיחה אפשריים* 2. *החולה בוחר את התפריט הרצוי והשימושי לאותה שיחה* 3. *המערכת מציגה את פרטי התפריט בצורה ברורה* 4. *החולה בוחר משפטים כרצונו, ונעזר באיורים כדי להבין את משמעותם.* 5. *המערכת מקריאה את המשפטים ומציגה אותם על גבי המסך בצורה ברורה ובהירה לקריאה והבנה* |
| הרחבות (שגיאות) | 1. *המשפט אותו מחפש החולה איננו נמצא במערכת*    1. *תתאפשר הזנת טקסט חופשי במהירות על המקום?* 2. *האדם הזר איננו מצליח לקרוא את המשפט שנבחר*    1. *המשפט מוצג בפונט קטן, המשפט לא ברור לקריאה בתאורה חזקה* 3. *המערכת לא משמיעה את המשפט ברמקולים*    1. *קביעת הרמקול הפנימי של המכשיר למצב עוצמה גבוהה בעת כניסה למערכת* |
| תרחישים חלופיים | *2. החולה משתמש בפונקציית החיפוש הקיימת באפליקציה למצוא את המשפט שברצונו להקריא בפני האדם זר* |
| שם התרחיש | *UC2. הוספה ידנית של ביטויים ומילים ע"י המשתמש* |
| שחקו ראשי | *חולה אפזיה* |
| מטרה | *הוספת מילים וביטויים לקטגוריה בהם יוכל להשתמש לצורך שיחות עתידיות, מילים וביטויים שנמצאות בשימוש ע"י המשתמש ואינן קיימות במערכת.* |
| היקף ורמה | *מערכת האפליקציה, המשתמש* |
| בעלי עניין ואינטרסים | *חולה האפזיה המעוניין להוסיף פריטים ללוח התקשורת שלו, קלינאית התקשורת המעוניינת שהאפליקציה תהווה כלי שימושי לחולה ביומיום.* |
| תיאור | *המשתמש מעוניין להוסיף לקטגוריה כלשהי מילים וביטויים בהם הוא מעוניין להשתמש לצורך תקשורת עם הסביבה. הוא לוחץ על הכפתור "הוספת פריט" המופיעה בחלק העליון של הקטגוריה בה הוא רוצה שהפריט ימצא.* |
| טריגר | *החולה נכנס למערכת באמצעות המכשיר הסלולרי.* |
| תנאי קדם | *החולה עבר את מסך ההזדהות (login) ומעוניין להוסיף פריט לקטגוריה קיימת.* |
| תנאי סיום מוצלח | *החולה הצליח להוסיף את הביטוי במלואו כולל תמונה מתאימה, האפליקציה מקריאה את הביטוי באופן נכון.* |
| תנאי סיום כישלון | *האפליקציה נכשלת בהוספת הפריט, לא הותאמה תמונה נכונה לפריט, האפליקציה מקריאה את שם הפריט באופן שגוי.* |
| **תרחיש הצלחה עיקרי** | 1. *החולה נכנס לקטגוריה בה הוא מעוניין להוסיף ביטוי חדש* 2. *החולה לוחץ על הכפתור "הוסף פריט"* 3. *החולה ממלא את שם הפריט ולוחץ על כפתור "האזן" כדי לוודא ששם הפריט לא נאמר באופן שגוי* 4. *החולה מוסיף תמונה התואמת את הפריט ע"י צילום, העלאה מהגלריה או חיפוש בגוגל.* 5. *החולה לוחץ על כפתור אישור.* 6. *הביטוי נשמר במערכת ומופיעה בקטגוריה המתאימה באופן מלא, כולל השם והתמונה שהוצמדו לו ע"י המשתמש* |
| הרחבות (שגיאות) | *1. האפליקציה קוראת את שם הפריט באופן שגוי*  *a. תתאפשר אפשרות לניקוד הטקסט כך שהאפליקציה תקריא באופן נכון*  *2. התמונה שהחולה בחר לא תואמת את שם הפריט*  *a. תתאפשר האופציה לעריכת הפריט לאחר הוספתו כולל שינוי שמו והתמונה המצורפת אליו.*  *3. החולה הוסיף פריטים בטעות, ישנם פריטים שברצונו למחוק*  *a. תתאפשר האופצייה למחוק את פריטים שנוספו ע"י המשתמש עצמו.* |
| תרחישים חלופיים | *4. החולה לא מעוניין להוסיף תמונה לפריט שאותו הוא מוסיף.* |

# סיפורי משתמשים – UserStories

|  |  |  |
| --- | --- | --- |
|  | סיפור | תיאור קצר |
| US1 | הוספת קטגוריה | בתור משתמש באפליקציה חולה אפזיה  אני מעוניין להוסיף קטגוריה שלא נמצאת במאגר  כך שאוכל להוסיף ביטויים התואמים לה המשמשים אותי בחיי היומיום לתקשורת עם הסביבה |
| US2 | הקראת משפט | בתור משתמש באפליקציה חולה אפזיה  אני מעוניין שהאפליקציה תקריא משפט לבחירתי  כך שאוכל להביע את עצמי בעת שיחות וביצירת תקשורת עם הסביבה |
| US3 | הוספת תמונה לפריט חדש | בתור משתמש באפליקציה חולה אפזיה  אני מעוניין שהאפליקציה תאפשר לי להוסיף תמונה מתאימה לביטוי חדש  כך שאוכל להבין את משמעות המילה ע"י איור מותאם. |
| US4 | שימוש בשורת "חיפוש" | בתור משתמש באפליקציה חולה אפזיה  אני מעוניין בחיפוש אחר מילה או ביטוי ע"י הקלדה  כך שאוכל למצוא במהירות את הביטוי המבוקש |
| US5 | הוספת ביטוי | בתור משתמש באפליקציה חולה אפזיה  אני מעוניין להוסיף ביטוי שלא נמצא במאגר  כך שאוכל להשתמש בו בעת תקשורת עם אחרים |
| US6 | התחברות לאפליקציה | בתור חולה אפזיה בעל משתמש קיים באפליקציה  אני מעוניין להתחבר לאפליקציה דרך המכשיר שברשותי (סלולרי/טאבלט/מחשב)  כך שאוכל להשתמש באפליקציה בעת תקשורת עם אחרים |
| US7 | הרשמה לאפליקציה | בתור חולה אפזיה  אני מעוניין להרשם לאפליקציה  כך שאוכל ליצור לוח תקשורת אישי בו אוכל להשתמש בתקשורת עם אחרים במהלך היומיום. |
| US8 | התנתקות מהאפליקציה | בתור חולה אפזיה בעל משתמש מחובר לאפליקציה  אני מעוניין באפשרות להתנתק מהאפליקציה |
| US9 | אחזור סיסמא | בתור חולה אפזיה בעל משתמש קיים באפליקציה  אני מעוניין באפשרות לאחזר את סיסמת המשתמש שלי  כך שאוכל להתחבר לאפליקציה גם אם שכחתי את הסיסמא. |
| US10 | מחיקת משתמש | בתור חולה אפזיה בעל משתמש קיים באפליקציה  אני מעוניין באפשרות למחוק את המשתמש  כך שאם לא ארצה לבצע בה שימוש נוסף כל פרטיי ימחקו מהמערכת. |
| US11 | אבטחה | בתור חולה אפזיה בעל משתמש קיים באפליקציה  אני מעוניין שהאפליקציה תהיה מאובטחת  כך שלאף אחד לא יהיה גישה לפרטיי האישיים השמורים בDATA BASE של האפליקציה. |
| US12 | מחיקת ביטוי (פריט) | בתור משתמש באפליקציה חולה אפזיה  אני מעוניין להוסיף ביטוי שלא נמצא במאגר  כך שאוכל להשתמש בו בעת תקשורת עם אחרים |

# דרישות סביבה

דרישות אלו מתארות את הסביבה הרחבה יותר של המוצר, חומרה, תוכנה, נתונים (בסיסי נתונים) וכדו'...



## דרישות חומרה

* מכשיר חכם (Smartphone) או מחשב (PC) או טאבלט
* גישה לרשת האינטרנט ואפשרות לגלישה בדפדפן.

## דרישות תוכנה

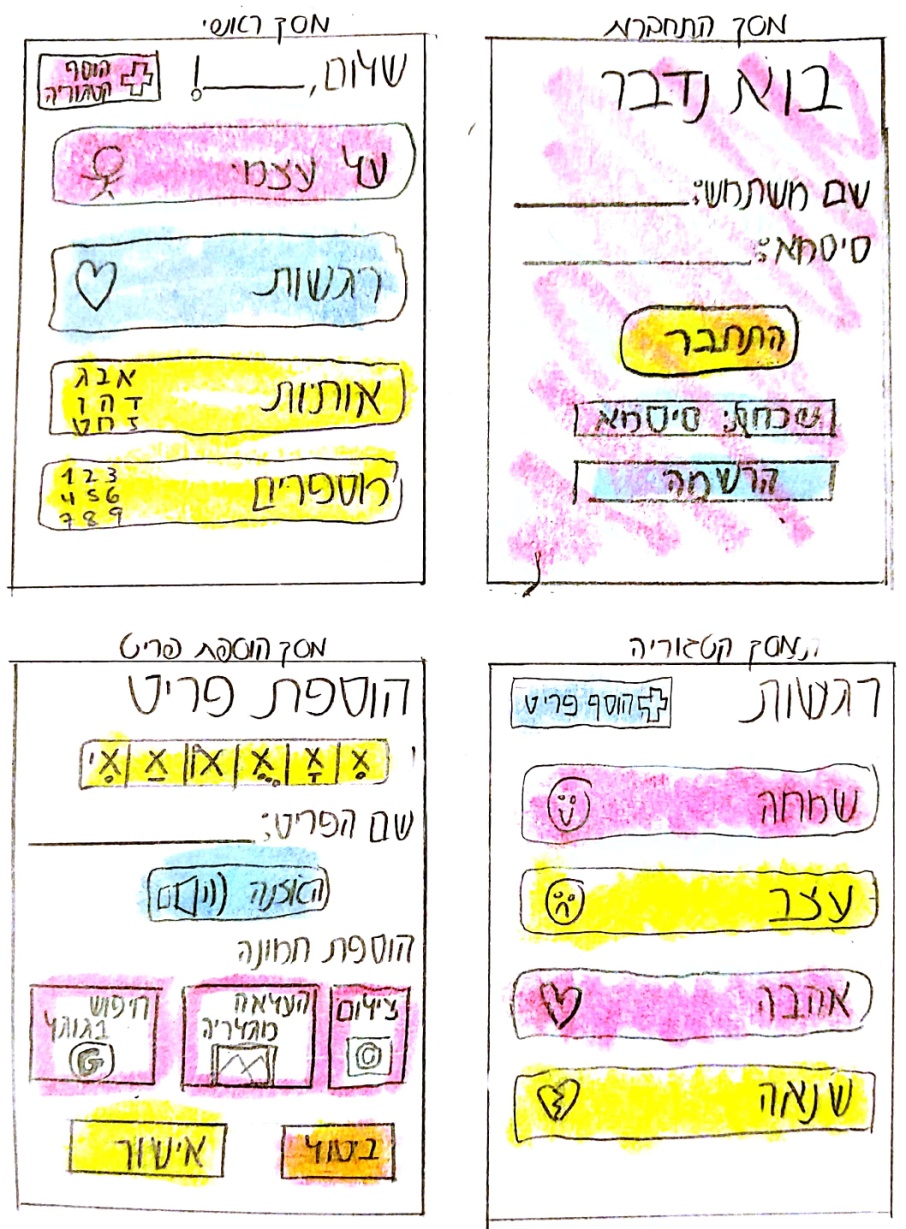
* Text to speech module – pjTextToSpeech / CodePan (<https://codepen.io/sunnygleason/pen/jrwEXy>)
* מסד נתונים (Database) המאוחסן על חוות השרתים של גוגל דרך Firebase.
* שימוש ב-Angular 2 לפיתוח המוצר.
* jQuery module – לעיצוב אפליקציית ה-web.

## דרישות נוספות

1. אבטחה
   * לכל חולה משתמש אישי משלו מוגן בסיסמה. (נשמר בזכרון – נכתב בפעם הראשונה בלבד כדי להקל על בעיות הזיכרון והכתיבה).
   * הסיסמה תישמר במסד הנתונים באמצעות SHA-1 כך שלא יהיה ניתן לקרוא אותה.
   * הפרטים האישיים אותם המשתמש הזין בעת ההרשמה וכמו כן לוחות התקשורת אותן יצר יהיו מאובטחים ולא תינתן גישה לפרטים אלו לגורמים חיצוניים.
2. נוחות
   * הטבלאות יהיו קריאות בכתב גדול ונגישות עם איורים רלוונטיים.
   * תינתן אפשרות לשנות את גודל התצוגה.
   * תינתן גלילה מטה לשם הצגת כלל לוחות התקשורת.
   * יצירת לוחות תקשורת אינדיבידואליים לכל משתמש.
   * הלוחות של כל משתמש ישמרו לו במסד הנתונים של האפליקציה תחת המשתמש הפרטי שלו.
   * לוחות בסיסיים משותפים לכלל המשתמשים כגון:
     1. "הצג את עצמי".
     2. "כתובת הבית שלי".
     3. "מספר הפלאפון שלי".

## ממשק משתמש – אב טיפוס

אב טיפוס לממשק המשתמש



# רשימת דרישות לבירור נוסף

* האם לכל משתמש יהיה יוזר אישי עם סיסמה? הרי חולי אפזיה מתקשים לזכור לכתוב ולקרוא כך שהם יתקשו בכתיבת היוזר ולזכור את הסיסמה. – ייתכן שישמר ע"י אדם קרוב אחר או ע"י קלינאית תקשורת?
* מעצב גרפי לשם האיורים – ישנו צורך ממשי וחשוב לאיורים. מהיכן ניתן להשיג את האיורים הרלוונטיים? האם ניתן להשיג מעצב גרפי?

תבנית סיכום סקר דרישות תוכנה - SRS Review

 תעדו את מהלך הסקר. את טבלת המשימות נזין בהמשך באתר ניהול הפרויקט כ-issues.

# תוכן

|  |  |
| --- | --- |
| שם הפרויקט הנסקר |  |
| נציגי הלקוח (הסוקרים) |  |
| מועד ומקום הפגישה |  |
| שמות משתתפים |  |

# נקודות שעלו בסקר

|  |  |
| --- | --- |
| 1. | חסר תרחיש חלופי בתרחיש UC2 |
| 2. |  |
|  |  |
|  |  |
|  |  |

# טבלת משימות הנובעות מהסקר (אפשר לחלופין למלא במערכת המשימות github issues)

|  |  |  |
| --- | --- | --- |
| נושא | פעולה נדרשת | באחריות |
| תרחיש UC2 | הוספת תרחיש חלופי במקרה של הזנת פרטים שגויים | אלן טיורינג |
|  |  |  |
|  |  |  |

1. ראו <http://en.wikipedia.org/wiki/Use_case#Design_scopes> [↑](#footnote-ref-1)