Software Requirements Specification

Version 2.0

<<Annotated Version>>

June 15, 2020

Virtual Assistant System

Bakri Alkhateeb

Muhammed Alkhattab

Submitted in fulfillment

Of the requirements of

Software Engineering Course

# Table of Contents

[Table of Contents ii](#_Toc43133218)

[List of Figures ii](#_Toc43133219)

[1.0. Introduction 1](#_Toc43133220)

[1.1. Purpose 1](#_Toc43133221)

[1.2. Scope of Project 1](#_Toc43133222)

[1.3. Glossary 2](#_Toc43133223)

[1.4. References 2](#_Toc43133224)

[1.5. Overview of Document 2](#_Toc43133225)

[2.0. Overall Description 3](#_Toc43133226)

[2.1 System Environment 3](#_Toc43133227)

[2.2 Functional Requirements Specification 4](#_Toc43133228)

[2.2.1 Use case: Fill Personal Information 4](#_Toc43133229)

[2.2.2 Use case: Voice Input 5](#_Toc43133230)

[2.2.3 **Use case: Alarm Queries** 6](#_Toc43133231)

[2.2.5 Search Web 7](#_Toc43133232)

[2.3 User Characteristics 7](#_Toc43133233)

[2.4 Non-Functional Requirements 7](#_Toc43133234)

[3.0. Requirements Specification 8](#_Toc43133235)

[3.1 Functional Requirements 8](#_Toc43133236)

[3.2.1 Fill Personal Information 8](#_Toc43133237)

[3.2.2 Voice Input 8](#_Toc43133238)

[3.2.3 Alarm Queries 9](#_Toc43133239)

[3.2.4 Map Queries 9](#_Toc43133240)

[3.2.5 Search Web 9](#_Toc43133241)

[3.3 Detailed Non-Functional Requirements 10](#_Toc43133242)

[3.3.1 Logical Structure of the Data 10](#_Toc43133243)

[3.3.2 Security 10](#_Toc43133244)

# List of Figures

[Figure 1 - System Environment 3](#_Toc77487669)

[Figure 2 – Fill Personal Information Process 4](#_Toc77487670)

[Figure 3 – Voice Input Process 5](#_Toc77487670)

[Figure 4 - Editor Use Cases 8](#_Toc77487671)

[Figure 5 - Logical Structure of the Article Manager Data 23](#_Toc77487672)

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of the Virtual Assistant System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to Mrs. Nahla Saad Eldeen for its approval.

## 1.2. Scope of Project

This software system will be a Virtual Assistant System for any user on a mobile phone. This system will be designed to maximize the user’s productivity by providing tools to assist in accomplishing daily tasks, which would otherwise have to be performed manually. By maximizing the user’s work efficiency and production the system will meet the user’s needs while remaining easy to understand and use.

## 1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Database | Collection of all the information monitored by this system. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| User | Mobile phone owner |

## 1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.5. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# 2.0. Overall Description

## 2.1 System Environment

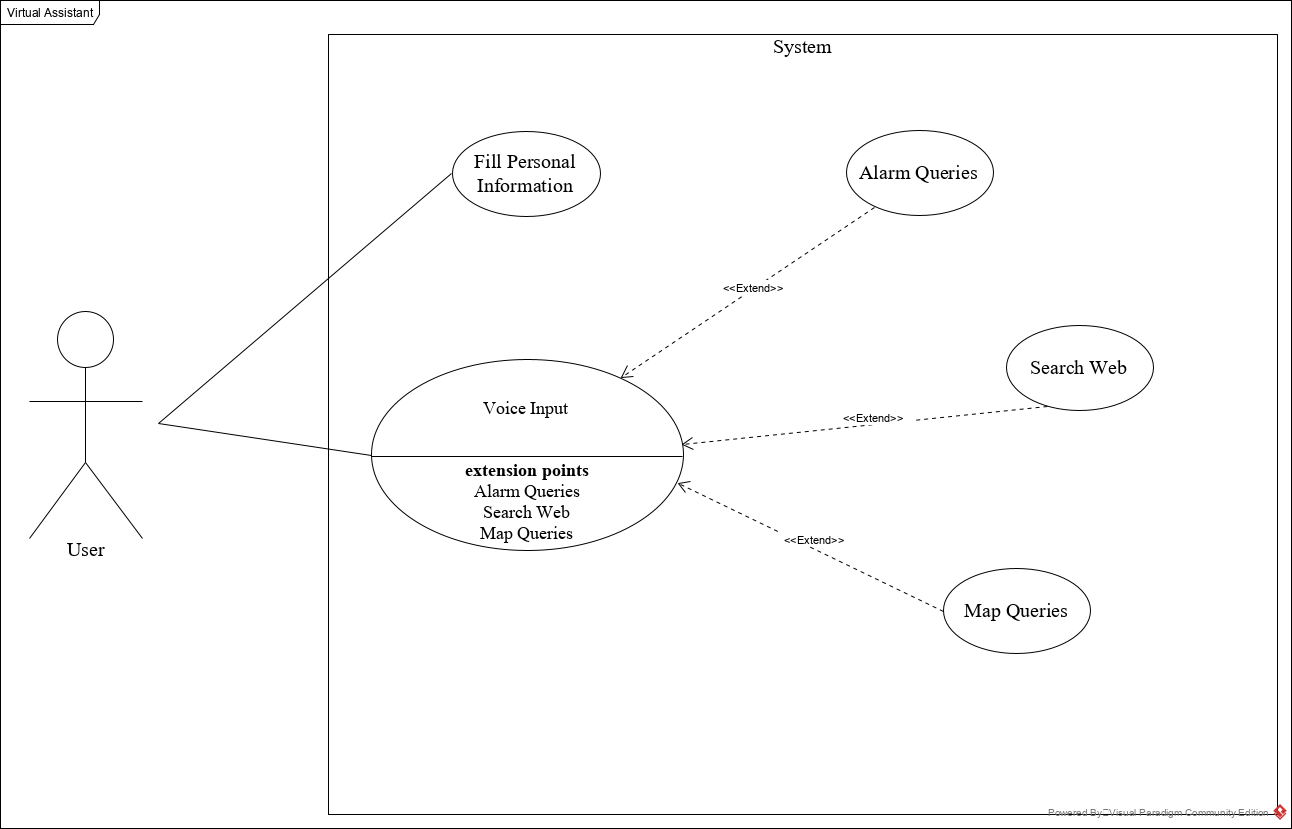


Figure - System Environment

The Virtual Assistant System has one active actor (the user).

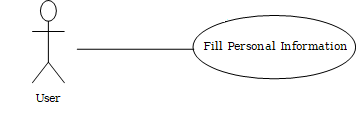
The user accesses the system through the application. Users communication with the system is through voice input.

## 2.2 Functional Requirements Specification

This section outlines the use cases for the user.

### 2.2.1 Use case: Fill Personal Information

**Diagram:**

****

**Brief Description**

The User accesses the system, inputs his personal information (e.g. name, age, job, …).

**Initial Step-By-Step Description**

Before this use case can be initiated, the User has already downloaded the Virtual Assistant application.

1. The User opens the application.
2. The system displays the information needed to be input.
3. The User inputs his/her personal information.
4. The system then transfers the user to the main application page.

**Xref:** Section 3.2.1, Fill Personal Information

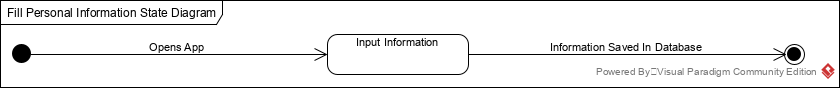
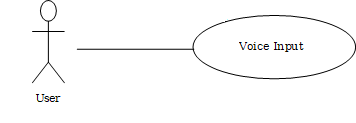


Figure - Fill Personal Information Process

The User opens the application after downloading and installing it. The system opens a bunch of pages requiring the user to fill some information about him/her so his experience gets to its best. The information filled then gets saved in the database to be used later in optimizing the user’s experience.

### 2.2.2 Use case: Voice Input

**Diagram:**

****

**Brief Description**

The User can use a keyword to activate the input functionality in the app.

**Initial Step-By-Step Description**

Before this use case can be initiated, the system has already activated the listener for the keyword.

1. The User says “Ok Assistant”.
2. The System starts listening to the user’s input.
3. The System then compiles the user’s input into commands to perform certain actions.

**Xref:** Section 3.2.2, Voice Input

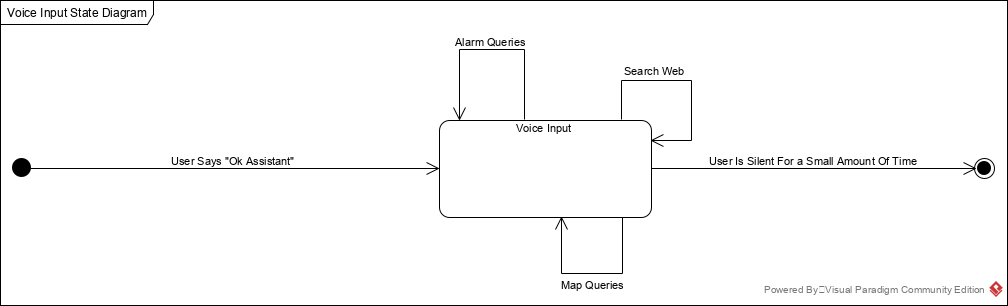


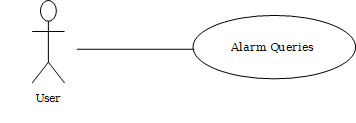
Figure 3 – Voice Input Process

The *Voice Input Process* state-transition diagram summarizes the use cases listed below.

The User starts off by saying “Ok Assistant”. The system then starts listening to the user voice commands. The User either makes Alarm Queries, Map Queries or Searches the web via voice commands. After a short amount of silence, the system stops listening to voice input and starts working on the commands input by the user.

### 2.2.3 **Use case: Alarm Queries**

**Diagram:**

****

**Brief Description**

The User tells the system to set an alarm for a certain time.

**Initial Step-By-Step Description**

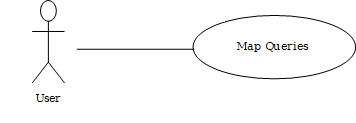
Before this use case can be initiated, the User has already said “Ok Assistant”.

1. The User makes an alarm query (e.g. wake me up in 45 minutes, set alarm on Sunday at 8:00 am).
2. The System uses the *voice input* to set an alarm for the user.

**Xref:** Section 3.2.3, Alarm Queries

2.2.4 Map Queries

**Diagram**



**Brief Description**

The User tells the system to set a waypoint or to look for a place on the map.

**Initial Step-By-Step Description**

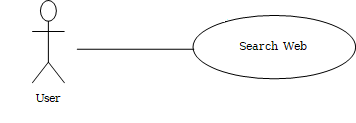
Before this use case can be initiated, the User has already said “Ok Assistant”.

1. The User makes a map query (e.g. find the shortest route to work, tell me where is the New York street is).
2. The System uses the *voice input* to set process map queries.

**Xref:** Section 3.2.4, Map Queries

### 2.2.5 Search Web

**Diagram**



**Brief Description**

The User tells the system to search the web for a certain information.

**Initial Step-By-Step Description**

Before this use case can be initiated, the User has already said “Ok Assistant”.

1. The User *tells* the system to look for something on the web (e.g. how to tie a tie,).
2. The System uses the *voice input* to search the web for the wanted information.

**Xref:** Section 3.2.5, Search Web

## 2.3 User Characteristics

The User is expected to be fluent in English and knows how to handle mobile phones.

## 2.4 Non-Functional Requirements

The system compiler will be on a server with high speed Internet capability. The physical machine to be used will be a mobile phone with internet connection.. The speed of the User’s connection will depend on the hardware used rather than characteristics of this system.

# 3.0. Requirements Specification

## 3.1 Functional Requirements

### Fill Personal Information

|  |  |
| --- | --- |
| **Use Case Name** | Fill Personal Information |
| **Xref** | Section 2.2.1 Fill Personal Information . |
| **Trigger** | The User opens the Virtual Assistant application |
| **Precondition** | The application is downloaded and installed |
| **Basic Path** | 1. The User opens the application. 2. The system displays the information needed to be input. 3. The User inputs his/her personal information. 4. The system then transfers the user to the main application page. |
| **Alternative Paths** | No alternative paths. |
| **Postcondition** | The User’s information is stored in the database. |
| **Exception Paths** | No exception paths. |
| **Other** | None |

### 3.2.2 Voice Input

|  |  |
| --- | --- |
| **Use Case Name** | Voice Input |
| **XRef** | Section 2.2.2 Voice Input. |
| **Trigger** | The User says the keyword “Ok Assistant”. |
| **Precondition** | The User has already filled his/her personal information. |
| **Basic Path** | 1. The User says “Ok Assistant”. 2. The System starts listening to the user’s input. |
| **Alternative Paths** | No alternative paths. |
| **Postcondition** | The System then compiles the user’s input into commands to perform certain actions. |
| **Exception Paths** | No exception paths. |
| **Other** | None |

### 3.2.3 Alarm Queries

|  |  |
| --- | --- |
| **Use Case Name** | Alarm Queries |
| **XRef** | Section 2.2.3 Alarm Queries . |
| **Trigger** | The User inputs a voice command related to alarms. |
| **Precondition** | The User says the keyword “Ok Assistant”. |
| **Basic Path** | The User makes an alarm query (e.g. wake me up in 45 minutes, set alarm on Sunday at 8:00 am). |
| **Alternative Paths** | No alternative paths. |
| **Postcondition** | The System uses the *voice input* to set an alarm for the user. |
| **Exception Paths** | No exception paths. |
| **Other** | None |

### 3.2.4 Map Queries

|  |  |
| --- | --- |
| **Use Case Name** | Map Queries |
| **XRef** | Section 2.2.4 Map Queries . |
| **Trigger** | The User inputs a voice command related to maps and routes. |
| **Precondition** | The User says the keyword “Ok Assistant”. |
| **Basic Path** | The User makes a map query (e.g. find the shortest route to work, tell me where is the New York street is). |
| **Alternative Paths** | No alternative paths. |
| **Postcondition** | The System uses the *voice input* to set process map queries. |
| **Exception Paths** | No exception paths. |
| **Other** | None |

### 3.2.5 Search Web

|  |  |
| --- | --- |
| **Use Case Name** | Search Web |
| **XRef** | Sec 2.2.5 Search Web. |
| **Trigger** | The User inputs a voice command related to web search and information not on the device. |
| **Precondition** | The User says the keyword “Ok Assistant”. |
| **Basic Path** | The User *tells* the system to look for something on the web (e.g. how to tie a tie,). |
| **Alternative Paths** | No alternative paths. |
| **Postcondition** | The System uses the *voice input* to search the web for the wanted information. |
| **Exception Paths** | No exception paths. |
| **Other** | None |

## 3.3 Detailed Non-Functional Requirements

### 3.3.1 Logical Structure of the Data

The logical structure of the data to be stored in the database is given below.

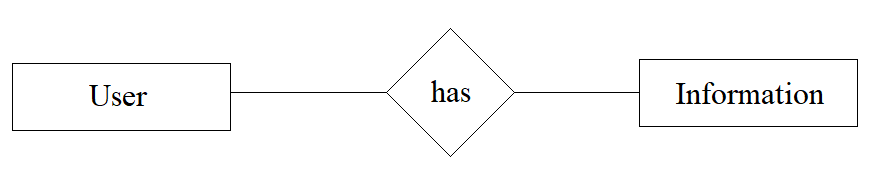


Figure - Logical Structure of the Data

The data descriptions of each of these data entities is as follows:

**User Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| First Name | Text | First Name of the user |  |
| Last Name | Text | Last Name of the user |  |
| Email Address | Text | User E-mail Address |  |
| Age | Integer | User Age |  |
| Work | Text | User Current Work |  |
| Phone Number | Integer | User Phone Number |  |
| Address | Text | User Address |  |

### 3.3.2 Security

The server on which the database resides will have its own security to prevent unauthorized access.

The Mobile on which the application resides will have its own security. Only the User will have physical access to the program on it.