



Software requirement specification document for Advanced Software(CSE232) project

Aley Amin
Mazen Sameh
Hassan Walid
Seif Elden

Supervised by: Dr. Ayman Ezzat, Eng. Raghda Ali

March 5, 2024

1 Introduction

1.1 Purpose of this document

This Software Requirements Specification (SRS) is a guide for building a website for a charity organization named Basmah. It outlines what the website needs to do and how it should work. This helps everyone involved in the project stay on the same page and makes sure the website meets the charity's needs.

1.2 Scope of this document

The project aims to establish Basmah's online presence, facilitate secure online donations, disseminate information about its initiatives, and enhance user engagement. Key stakeholders include the charity organization, donors, web developers, and system engineers. Constraints encompass predefined schedules, budget limitations, adherence to software engineering standards, and regulatory compliance. This document serves as a comprehensive guide to align project participants, ensuring a shared understanding of objectives and requirements, subject to periodic review for updates.

1.3 Overview

The product defined as a result of the outlined vision and mission is a comprehensive educational and ethical development program aimed at children. This program seeks to address the challenges of corruption and the lack of education, fostering a new generation with a strong moral compass and a deep sense of responsibility towards their community.

1.4 Business Context

The program is designed to provide children with the necessary tools to overcome the barriers imposed by corruption and incomplete education, enabling them to pursue their dreams with confidence and determination. It includes elements such as mentorship, skill development, and character-building activities to shape well-rounded individuals. elicitation process.

2 General Description

2.1 Product Functions

the product should let a user apply for one of the 4 different paths provided, book appointment for an interview, and donate for his desired path.

2.2 Similar System Information

The system would in concept be similar to any other charitable organization eg:- resala but smaller in size compared to them as the system will be tailored for the needs of basmah.

2.3 User Characteristics

Describes the features of the user community, including their expected expertise with software systems and the application domain.

2.4 User Problem Statement

This section describes the essential problem(s) currently confronted by the user community.

2.5 User Objectives

This section describes the set of objectives and requirements for the system from the user's perspective. It may include a "wish list" of desirable characteristics, along with more feasible solutions that are in line with the business objectives.

2.6 General Constraints

Lists general constraints placed upon the design team, including speed requirements, industry protocols, hardware platforms, and so forth.

3 Functional Requirements

This section lists the functional requirements in ranked order. Functional requirements describes the possible effects of a software system, in other words, what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements see table 25. Each functional requirement should be specified in a format similar to the following:

Table 1: Restrict User Personal Info Access Function Description

Function Name	restrictUserPersonalInfoAccess
Description	The function restrictUserPersonalInfoAccess is designed to enhance user privacy and security within a system. Its primary purpose is to control and limit access to sensitive personal information, the function checks the user's access level, each level is able to view certain information.
Inputs	User credentials (AccessLevel).
Source	User try to access another user information.
Outputs	Alert message.
Destination	User interface.
Action	Verify user credentials against stored data.
Requirements	Multiple access levels, Each user must have only one level.
Criticality	Essential for system security and data integrity.
Side effects	None
Risks	User authentication failures, potential security breaches.
Pre-Condition	Low level user attempting to access high level information.
Post-Condition	User granted access or denied with appropriate feedback.

Table 2: Alert System Failure Function Description

Function Name	AlertSystemFailure
Description	This function is a way for the system to achieve availability. the function is called upon failure of the website to respond, a message is then sent to the system admins to alert them about this failure, and a message to users.
Inputs	System failure (responseError).
Source	System is unresponsive.
Outputs	Alert message.
Destination	Admin contact.
Action	Alert admins of system failure, and inform users of downtime.
Requirements	Tracking system response.
Criticality	Essential for system availability.
Side effects	None
Risks	Response tracking failure, failure to send message to admins.
Pre-Condition	System response failure.
Post-Condition	Alert message to admins and downtime message

Table 3: List all donation causes Function description

Function Name	ListDonationCauses
Description	The function shows all the available causes of donation. It exists at the Donations page. the causes listed are (Janeen, Bareeq, Elaf, Marsah)
Inputs	Event Listener for Donations button
Source	User presses on donate button.
Outputs	View donation causes
Destination	User interface.
Action	Open Donations page
Requirements	None
Criticality	High
Side effects	None
Risks	failure to open donations page
Pre-Condition	user at home page.
Post-Condition	User at Donations page.

Table 4: Select donation cause Function description

Function Name	selectDonationCause
Description	The function takes the choice of the user on which cause he wants to donate to and sends them to the payment page.
Inputs	Event Listener for Donation cause button
Source	User presses on one of the donation cause buttons.
Outputs	View payment page
Destination	User interface.
Action	Open payment page
Requirements	None
Criticality	High
Side effects	None
Risks	failure to open payment page
Pre-Condition	user at donations page.
Post-Condition	User at payment page.

Table 5: List all donation types Function description

Function Name	ListDonationTypes
Description	The function shows all the available types of donation. It exists at the Donations page. The types of donations are (money, clothes, food)
Inputs	Event Listener for Donations button
Source	User presses on donate button.
Outputs	View donation type
Destination	User interface.
Action	Open Donations page
Requirements	None
Criticality	High
Side effects	None
Risks	failure to open donations page
Pre-Condition	user at home page.
Post-Condition	User at Donations page.

Table 6: Select donation type Function description

Function Name	selectDonationType
Description	The function takes the choice of the user on what he wants to donate. if it is money, send them to the payment page.
Inputs	Event Listener for Donation type button
Source	User presses on one of the donation type buttons.
Outputs	View payment page or send email to admin about type of donation
Destination	User interface.
Action	Open payment page if the chosen type of donation is money. if the type of donation is something else then an email is sent to the admin informing them that a person is donating a physical thing.
Requirements	None
Criticality	High
Side effects	None
Risks	failure to open payment page, failure to send email
Pre-Condition	user at donations page.
Post-Condition	User at payment page.

Table 7: System Calculates Money Function Description

Function Name	calculatateTotalDonations
Description	The function calculatateTotalDonations takes the donation cause and amount to be donated ,and calculates the total. It will display the totals in the payment page.
Inputs	Donation Information (DonationCause , Donation amount).
Source	User donates money through the payment page.
Outputs	Total amount of money donated
Destination	User interface.
Action	Calculate the total amount of money donated till last transaction.
Requirements	None
Criticality	High.
Side effects	None
Risks	Wrong amount calculated .
Pre-Condition	User chooses donation cause and amount they want to donate.
Post-Condition	Calculated amount is displayed at payment page.

Table 8: List Payment Method Function Description

Function Name	listPaymentMethod
Description	The function listPaymentMethod shows the different payment methods at payment page.
Inputs	Event Listener for method button.
Source	User presses on payment methods button.
Outputs	List of payment methods.
Destination	User interface.
Action	Display all available payment methodes
Requirements	amount of total donations is not smaller than 10L.E
Criticality	High
Side effects	None
Risks	No payment method is available
Pre-Condition	payment page without list of payment methods
Post-Condition	appearance of payment methods.

Table 9: Choose Payment Method Function Description

Function Name	choosePaymentMethod
Description	The function choosePaymentMethod takes the payment method chosen by the user.
Inputs	Event Listener for methodType button, donation amount
Source	User presses on a method button.
Outputs	payment information dialog.
Destination	User interface.
Action	Display a dialog to display the instapay address, bank and account for the instapay method. Display dialog with vodafone cash payment information for the vodaphone cash method. Display a dialog to enter card information for the Card method.
Requirements	Service for the chosen payment method is available
Criticality	High
Side effects	None
Risks	No payment method is available
Pre-Condition	Donation amount entered
Post-Condition	appearance of the dialog

Table 10: Confirm Payment Function Description

Function Name	confirmPayment
Description	The function confirmPaymen displays a popup for the user to confirm the amount of money donated.
Inputs	Event Listener for Confirmation button
Source	user enters card information and presses the confirmation button
Outputs	popup alerting the success of the transaction.
Destination	User interface.
Action	the user enters his card information which is followed by a popup asking for the confirmation of the user to this transaction. upon pressing the confirmation button a new popup appears alerting of the success or failure of the transaction
Requirements	card information is entered
Criticality	High
Side effects	None
Risks	wrong card information, failure of transaction, popup does not appear
Pre-Condition	payment information dialog
Post-Condition	success or failure of transaction

Table 11: List Available volunteering paths Function description

Function Name	ListPaths
Description	This function shows the user the different paths offered for volunteering by Basmah. .
Inputs	Event Listener for ViewPaths button
Source	User presses on view paths button.
Outputs	View Paths Page.
Destination	User interface.
Action	Open view paths page
Requirements	existence of volunteering paths.
Criticality	High
Side effects	None
Risks	failure to open view paths page.
Pre-Condition	user at home page.
Post-Condition	User at volunteering page.

Table 12: Select volunteering path Function description

Function Name	selectVolunteeringPath
Description	The function takes the choice of the user on which path he wants to volunteer to and sends them to the appointments page.
Inputs	Event Listener for volunteering path button
Source	User presses on one of the volunteering path buttons.
Outputs	View appointments page
Destination	User interface.
Action	Open appointment page
Requirements	None
Criticality	High
Side effects	None
Risks	failure to open appointment page
Pre-Condition	user at volunteering page.
Post-Condition	User at appointment page.

Table 13: List all appointment days Function Description

Function Name	listAppointmentDays
Description	The function shows a list of available days for an appointment for the chosen volunteering path, this list is at the appointments page.
Inputs	user's volunteering choice
Source	appointment page is opened
Outputs	list of all available days for appointments
Destination	User interface.
Action	the function takes the user's choice on which path he wants to volunteer to and shows a list of all available days for an appointment to this path.
Requirements	volunteering path has appointment days
Criticality	High
Side effects	None
Risks	wrong list of days.
Pre-Condition	empty appointment page
Post-Condition	appointment page with list of days

Table 14: Book appointment Function Description

Function Name	bookAppoointment
Description	The function takes the day choosen by the user for an appointment.
Inputs	Event Listener for appointment day button
Source	user presses a day button
Outputs	popup alerting the user of his chosen day with a confirmation button.
Destination	User interface.
Action	the user chooses a day from the list of appointment days, upon choosing a popup appears with the chosen day and a confirmation button
Requirements	a day is chosen from the list
Criticality	High
Side effects	None
Risks	None
Pre-Condition	a day is chosen
Post-Condition	appearance of popup

Table 15: Confirm appointment Function Description

Function Name	confirmAppointment
Description	The function confirmAppointment takes the chosen day by the user and sends it to the admin.
Inputs	Event Listener for Confirmation button
Source	user presses on confirmation button
Outputs	popup alerting the user that his appointment is being reviewed by an admin, email to admin about the appointment information
Destination	User interface.
Action	the user presses on the confirmation button. upon pressing the function sends an email to the admin about the appointment and a popup for the user alerting him that his appointment is being reviewed by an admin
Requirements	None
Criticality	High
Side effects	None
Risks	popup doesn't appear. email is not sent to admin.
Pre-Condition	popup of appointment confirmation
Post-Condition	popup of appointment being reviewed by admin

Table 16: Cancel appointment Function Description

Function Name	cancelAppointment
Description	The function cancelAppointment sends an email to the admin informing him of the cancelation of the appointment.
Inputs	Event Listener for Cancel button
Source	user presses on cancel button
Outputs	popup alerting the user that his appointment is canceled
Destination	User interface.
Action	the user presses on the cancel button. upon pressing the function sends an email to the admin about the appointment being canceled and a popup for the user alerting him that his appointment is canceled
Requirements	None
Criticality	High
Side effects	None
Risks	popup doesn't appear. email is not sent to admin.
Pre-Condition	cancel button at appointment page
Post-Condition	popup of appointment cancelation

Table 17: Add user Function Description

Function Name	addUser
Description	The admin is able to add a user from the admin page.
Inputs	user information(Name,Age,Number.Email)
Source	admin adds a user through admin page
Outputs	A new user add to database
Destination	Users database.
Action	the admin enters the user information then presses on add user button. then a new user is add to the users database and a popup appears confirming that the user has been added.
Requirements	applicable user information
Criticality	medium
Side effects	None
Risks	user is not add to database
Pre-Condition	add user page
Post-Condition	popup confirming user is added successfully

Table 18: Delete user Function Description

Function Name	DeleteUser
Description	The admin is able to delete a user from the admin page.
Inputs	user information(Name,Age,Number.Email)
Source	admin deletes a user through admin page
Outputs	A user is deleted from database
Destination	Users database.
Action	the admin enters the user information then presses on delete user button. then the user is deleted from the users database if he exists and a popup appears confirming that the user has been deleted. if the user is not in the users database, a popup appears alerting the admin that the user does not exist
Requirements	user exists in database
Criticality	medium
Side effects	None
Risks	user is not deleted from database
Pre-Condition	delete user page
Post-Condition	popup confirming user is deleted successfully or does not exist

Table 19: Confirm volunteer appointment Function Description

Function Name	confirmVolunteerAppointment
Description	The admin confirms the volunteer's appointment after receiving the appointment email and an email is sent to the volunteer automatically and adds the appointment to the appointments database.
Inputs	volunteer information (Name, Age, Number, Email)
Source	admin recives an appointment email
Outputs	email sent to volunteer confirming his appointment and appointment add to database
Destination	volunteer's email
Action	the admin receives a volunteer's appointment email, the admin then confirms the appointment which calls this function sending an email automatically to the volunteer informing him that the appointment is confirmed and adds the appointment to the appointments database.
Requirements	Admin receives an appointment email
Criticality	High
Side effects	None
Risks	confirmation Eamil is not sent, appointment is not add to database
Pre-Condition	appointment email
Post-Condition	Confirmation email

Table 20: Add Admin Function Description

Function Name	addAdmin
Description	The admin is able to add an admin from the admin page.
Inputs	Admin information(Name,Age,Number.Email)
Source	admin adds an admin through admin page
Outputs	A new admin add to database
Destination	Admins database.
Action	the admin enters the new admins information then presses on add admin button. then a new admin is add to the admins database and a popup appears confirming that the admin has been added.
Requirements	applicable admin information
Criticality	medium
Side effects	None
Risks	admin is not add to database
Pre-Condition	add admin page
Post-Condition	popup confirming admin is added successfully

Table 21: View all appointments Function Description

Function Name	viewAppointments
Description	The function shows a all appointments confirmed by the admin through the view appointments page open only for admins.
Inputs	None
Source	view appointments page
Outputs	list of all appointments confirmed by the admin
Destination	admin page.
Action	the function takes all appointments inside the appointments database and shows them to the admin through the view appointments page.
Requirements	None
Criticality	Medium
Side effects	None
Risks	appointments do not appear.
Pre-Condition	empty view appointments page
Post-Condition	appointments page with list of all appointments confirmed

Table 22: View all users Function Description

Function Name	viewUsers
Description	The function shows a all users registered through view users page open only for admins.
Inputs	None
Source	view users page
Outputs	list of all users registered
Destination	admin page.
Action	the function takes all users inside the users database and shows them to the admin through the view users page.
Requirements	None
Criticality	Medium
Side effects	None
Risks	Users do not appear.
Pre-Condition	empty view users page
Post-Condition	Users page with list of all users registered

Table 23: Confirm physical donation Function Description

Function Name	confirmPhysicalDonation
Description	The admin confirms the physical donation (clothes,food) after receiving the donation email and an email is sent to the user automatically informing them that they will receive the donation.
Inputs	donation email
Source	admin receives a donation email
Outputs	email sent to user confirming his donation is appreciated and will be received
Destination	user's email
Action	the admin receives a donation email, the admin then confirms the donation which calls this function sending an email automatically to the user informing him that the donation is appreciated and will be received.
Requirements	Admin receives a donation email
Criticality	High
Side effects	None
Risks	confirmation Email is not sent.
Pre-Condition	donation email
Post-Condition	Confirmation email

Table 24: Register Function Description

Function Name	Register
Description	The function allows a user to register to the system.
Inputs	user information (Name, Age, Number, Email, Password)
Source	Sign in page
Outputs	A new user is add to the users database, a popup confirming success or failure of registration.
Destination	sign in page.
Action	the function takes the information entered by the user. if the information is applicable, a new user is add to the users database then a popup appears confirming a successful registration and redirects the user to sign in page, and if information is not applicable then a popup appears informing of the failure of registration.
Requirements	user enters information
Criticality	Low
Side effects	None
Risks	User is not add to Users database, popup doesn't appear.
Pre-Condition	user registers his account
Post-Condition	user redirected to sign in page if successful registration.

Table 25: SignIn Function Description

Function Name	SignIn
Description	The function signs in a user or an admin if they exist.
Inputs	information (Email,Password)
Source	Sign in page
Outputs	success or failure of signing in.
Destination	sign in page.
Action	the function takes the information entered by the user or admin. if they exist then sign in is successful if not then a popup appears informing them that they need to register first.
Requirements	information is entered
Criticality	Low
Side effects	None
Risks	Unable to sing in.
Pre-Condition	User or admin enters his account
Post-Condition	user redirected to home page if successful signin.

4 Interface Requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include library routines, token streams, shared memory, data streams, and so forth.

4.1 User Interfaces

Use some software for primitive plan of your project. Describes how this product interfaces with the user.

4.1.1 GUI

Describes the graphical user interface if present. This section should include a set of screen dumps or mock-ups to illustrate user interface features. If the system is menu-driven, a description of all menus and their components should be provided.

4.1.2 CLI

Describes the command-line interface if present. For each command, a description of all arguments and example values and invocations should be provided.

4.1.3 API

Describes the application programming interface, if present. For each public interface function, the name, arguments, return values, examples of invocation, and interactions with other functions should be provided.

4.1.4 Diagnostics or ROM

Describes how to obtain debugging information or other diagnostic data.

4.2 Hardware Interfaces

Describes interfaces to hardware devices.

4.3 Communications Interfaces

Describes network interfaces.

4.4 Software Interfaces

Describes any remaining software interfaces not included above.

5 Design Constraints

Specifies any constraints for the design team using this document.

5.1 Standards Compliance

5.2 Hardware Limitations

5.3 others as appropriate

6 Non-functional Requirements

6.1 Security

In order to uphold the highest standards of privacy and data security, access to personal information of individual users is restricted to designated administrators only. Users are expressly prohibited from viewing or accessing personal details of fellow users. This measure ensures the confidentiality and integrity of sensitive information. A functional requirement named `restrictUserPersonalInfoAccess(accessLevel)` that will check the user's access level to either hide other users' information or not.

6.2 Compatibility

The system should be compatible with different devices(mobile phone, computers, tablets). This will be implemented by insuring that bootstrap is used in all pages.

6.3 Accessibility

System should comply with disability regulations. This will be applied by integrating "text-to-speech".

6.4 Availability

The system should be accessible for as long as possible and upon failure messages should be sent ensuring that the staff are aware of this failure and measures are taken. This will be implemented by using the functional requirement `alertSystem-Failure(responseError)`.

7 Preliminary Object-Oriented Domain Analysis

This section presents a list of the fundamental objects that must be modeled within the system to satisfy its requirements. The purpose is to provide an alternative, "structural" view on the requirements stated above and how they might be satisfied in the system. A primitive class diagram to be delivered.

7.1 Inheritance Relationships

This section should contain a set of graphs that illustrate the primary inheritance hierarchy (is-kind-of) for the system. For example:

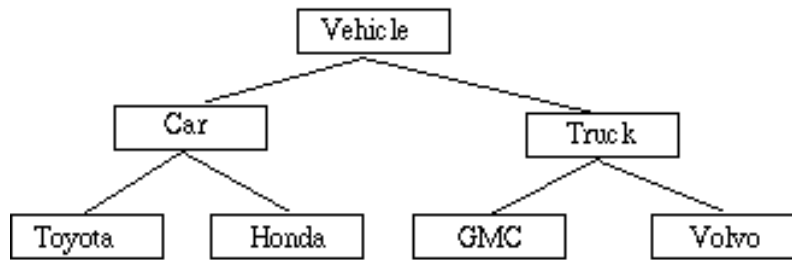


Figure 1: Inheritance Relations

7.2 Class descriptions

This section presents a more detailed description of each class identified during the OO Domain Analysis. For more details on the process giving rise to these descriptions, see Lecture 5.3: OO Domain Analysis and/or texts on object-oriented software development. Each class description should conform to the following structure:

7.2.1 Class name

Abstract or Concrete: Indicates whether this class is abstract or concrete.

7.2.2 List of Superclasses:

Names all immediate superclasses.

7.2.3 List of Subclasses:

Names all immediate subclasses.

7.2.4 Purpose:

States the basic purpose of the class.

7.2.5 Collaborations:

Names each class with which this class must interact in order to accomplish its purpose, and how.

7.2.6 Attributes:

Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).

7.2.7 Operations

: Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.

7.2.8 Constraints:

Lists any restrictions upon the general state or behavior of instances of this class.

8 Operational Scenarios

This section should describe a set of scenarios that illustrate, from the user's perspective, what will be experienced when utilizing the system under various situations. In the article Inquiry-Based Requirements Analysis (IEEE Software, March 1994), scenarios are defined as follows: In the broad sense, a scenario is simply a proposed specific use of the system. More specifically, a scenario is a description of one or more end-to-end transactions involving the required system and its environment. Scenarios can be documented in different ways, depending up on the level of detail needed. The simplest form is a use case, which consists merely of a short description with a number attached. More detailed forms are called scripts. These are usually represented as tables or diagrams and involved identifying an action and the agent (doer) of the action. For this reason, a script can also be called an action table. Although scenarios are useful in acquiring and validating requirements, they are not themselves requirements, because they describe the system's behavior only in specific situations; a specification, on the other hand, describes what the system should do in general.

9 Preliminary Schedule Adjusted

This section provides an initial version of the project plan, including the major tasks to be accomplished, their interdependence's, and their tentative start/stop dates. The plan also includes information on hardware, software, and resource requirements. The project plan should be accompanied by one or more PERT or GANTT charts.

10 Preliminary Budget Adjusted

This section provides an initial budget for the project, itemized by cost factor.

11 Appendices

Specifies other useful information for understanding the requirements. All SRS documents should include at least the following two appendices:

11.1 Definitions, Acronyms, Abbreviations

Provides definitions of unfamiliar definitions, terms, and acronyms.

11.2 Collected material

12 References

References