

<Project Title>

Software Requirements Specification (SRS)

Version 1.0

Group member	Registration number	Email
<Name 1>	<Reg. No.>	<email>
<Name 2>	<Reg. No.>	<email>
<Name 3>	<Reg. No.>	<email>

Item	Details
Supervisor	<Name, designation>
Co-supervisor	<Name, designation, if any>
Department	<Department of Computer Science / Software Engineering>
University	<University name, campus>
Programme and session	<BSCS / BSSE, session 20XX to 20XX>
Submission date	<DD Month YYYY>

Template from ezitech.org/srs-template-for-fyp. You may remove this line.

Revision history

Version	Date	Description of change	Changed by
1.0	<DD/MM/YYYY>	First draft submitted to supervisor	<Name>
1.1	<DD/MM/YYYY>	<What changed after review>	<Name>

Table of contents

[Guidance: In Word, right-click the table of contents and choose Update Field after you finish writing. Delete this note before submission.]

1. Introduction

1.1 Purpose

[Guidance: One paragraph. Say what this document specifies, which version of the system it covers, and who will read it (supervisor, evaluation committee, your own team). Delete this note before submission.]

<Write the purpose here>

1.2 Product scope

[Guidance: What the system is, what problem it solves, and for whom. State clearly what is in scope and what is out of scope. Keep this consistent with your approved proposal. Delete this note before submission.]

<Write the scope here>

1.3 Definitions, acronyms and abbreviations

Term	Meaning
SRS	Software Requirements Specification
FR	Functional requirement
NFR	Non-functional requirement
<Term>	<Meaning in this project>

1.4 References

[Guidance: List the documents and sources you relied on: your approved proposal, IEEE 830 or ISO/IEC/IEEE 29148, any API documentation, research papers. Use the citation style your department asks for. Delete this note before submission.]

- <Approved FYP proposal, version and date>
- <IEEE Std 830-1998, Recommended Practice for Software Requirements Specifications>
- <Other reference>

1.5 Document overview

[Guidance: Two or three sentences describing how the rest of this document is organised. Delete this note before submission.]

<Write the overview here>

2. Overall Description

2.1 Product perspective

[Guidance: Is this a new standalone system, a replacement for a manual process, or a module of a larger system? Mention any existing system it connects to. A simple block diagram helps. Delete this note before submission.]

<Write the product perspective here>

2.2 Product functions

[Guidance: A short summary of the main features, grouped by module. The detailed, numbered requirements come in Section 4. Delete this note before submission.]

- <Module 1: one line summary>
- <Module 2: one line summary>
- <Module 3: one line summary>

2.3 User classes and characteristics

User class	What they do in the system	Technical skill	How often
Admin	Manages users, settings and reports	Medium	Daily
<User class>	<Main tasks>	<Low / Medium / High>	<Daily / Weekly>

2.4 Operating environment

[Guidance: Where the system runs: web browsers supported, Android or iOS versions, server operating system, database, hosting. Give versions. Delete this note before submission.]

<Write the operating environment here>

2.5 Design and implementation constraints

[Guidance: Anything that limits your choices: required programming language or framework, university hosting, budget, offline use, Urdu language support, data privacy rules. Delete this note before submission.]

<Write the constraints here>

2.6 Assumptions and dependencies

[Guidance: Things you assume to be true and external services you depend on, for example a payment gateway sandbox, an SMS API, or users having smartphones with internet. Delete this note before submission.]

<Write the assumptions and dependencies here>

3. External Interface Requirements

3.1 User interfaces

[Guidance: Describe the main screens and paste low fidelity sketches or wireframes. Mention any standard you follow, such as responsive design or a style guide. Delete this note before submission.]

<Describe the user interfaces here>

3.2 Hardware interfaces

[Guidance: Only if the project uses hardware: sensors, Arduino or Raspberry Pi boards, cameras, barcode or fingerprint scanners. Otherwise write Not applicable. Delete this note before submission.]

<Describe hardware interfaces or write Not applicable>

3.3 Software interfaces

[Guidance: Third party APIs, libraries and services the system talks to, with the data exchanged. Delete this note before submission.]

Service or API	Purpose	Data exchanged
<Google Maps API>	<Show nearby locations>	<Coordinates in, map tiles out>
<Name>	<Purpose>	<Data>

3.4 Communication interfaces

[Guidance: Protocols and formats: HTTPS, REST with JSON, WebSockets, email or SMS notifications. Delete this note before submission.]

<Describe communication interfaces here>

4. Functional Requirements

[Guidance: Number every requirement (FR-01, FR-02 and so on) and group them by module. Write each as one testable sentence that starts with "The system shall". The grey rows are examples from a sample appointment booking project; replace them with your own. Delete this note before submission.]

4.1 <Module name>

ID	Requirement	Priority	User class
FR-01	The system shall allow a registered patient to log in with an email address and password.	High	Patient
FR-02	The system shall show a doctor's available time slots for the next 14 days.	High	Patient
FR-03	The system shall send an email confirmation within one minute of a booking.	Medium	Patient
FR-04	<The system shall ...>	<High / Medium / Low>	<User class>

4.2 <Module name>

ID	Requirement	Priority	User class
FR-05	<The system shall ...>	<Priority>	<User class>
FR-06	<The system shall ...>	<Priority>	<User class>

5. Use Cases

5.1 Use case diagram

[Guidance: Insert one diagram showing all actors and use cases. Draw it in draw.io, Lucidchart, StarUML or Visual Paradigm and paste it as an image. Delete this note before submission.]

<Insert the use case diagram here>

5.2 Use case descriptions

[Guidance: Copy this table once for every use case in the diagram. Delete this note before submission.]

Item	Details
Use case ID	UC-01
Use case name	<Book appointment>
Actor(s)	<Patient>
Related requirements	<FR-01, FR-02, FR-03>
Preconditions	<Patient is logged in>
Main flow	<1. Patient selects a doctor. 2. System shows free slots. 3. Patient picks a slot and confirms. 4. System saves the booking and sends a confirmation.>
Alternate flows	<2a. No free slots: system offers the next available date.>
Postconditions	<Booking is stored and the slot is no longer available>

6. Non-functional Requirements

[Guidance: Every non-functional requirement needs a number or a condition that can be checked. "The system shall be fast" cannot be tested. "Search results shall load within 2 seconds for up to 10,000 records" can. Delete this note before submission.]

ID	Category	Requirement	How it will be checked
NFR-01	Performance	Pages shall load within 3 seconds on a 4G connection.	Browser network timing on 10 page loads
NFR-02	Security	Passwords shall be stored as salted hashes, never in plain text.	Inspect the users table
NFR-03	Usability	A new user shall complete a booking in under 2 minutes without help.	Timed test with 5 users
NFR-04	Availability	<Requirement>	<Check>
NFR-05	Maintainability	<Requirement>	<Check>
NFR-06	Portability	<Requirement>	<Check>

7. Data Requirements

[Guidance: List the main entities the system stores and their key attributes. Put the full entity relationship diagram in your design document, or here if your department asks for it. Delete this note before submission.]

Entity	Key attributes	Related to
<User>	<user_id, name, email, role>	<Booking>
<Entity>	<Attributes>	<Entity>

8. Requirements Traceability Matrix

[Guidance: Link every requirement to the use case that covers it and the test case that verifies it. Fill the test case column after you write the test document. Supervisors use this table to check that nothing was dropped. Delete this note before submission.]

Requirement ID	Use case	Test case ID	Status
FR-01	UC-01	TC-01	<Pass / Fail / Pending>
FR-02	UC-01	TC-02	<Status>
<ID>	<UC>	<TC>	<Status>

Appendix A: Work Plan

[Guidance: Include this only if your department asks for a schedule in the SRS. Many attach a Gantt chart instead. Delete this note before submission.]

Task	Start	End	Owner
Requirements and SRS	<Week 1>	<Week 4>	<Name>
Design (diagrams, database)	<Week 5>	<Week 7>	<Name>
Development: <module>	<Week 8>	<Week 14>	<Name>
Testing and documentation	<Week 15>	<Week 18>	<All>

Appendix B: Approval

Name	Signature	Date
<Supervisor>		
<Co-supervisor>		