

SOFTWARE REQUIREMENTS SPECIFICATION

ENVISION

An internship hunting and project building website designed specifically for the student community. The goal of the site is to make a positive difference to the student, to the employer, and to the society at large.

Version 1.0

Prepared By
Yash Vakil

Revision History

Name	Date	Reason For Changes	Version

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Definition, Acronyms and Abbreviations	1
1.4 References	2
1.5 Overview	2
2. Overall description	3
2.1 Product perspective	3
2.2 Product architecture	3
2.3 Product functions.....	3
2.4 User characteristics.....	4
2.5 Constraints	4
2.6 Assumptions and dependencies	4
3. Specific requirements	5
3.1 External interface Requirements	5
3.1.1 User interfaces	5
3.1.2 Hardware interfaces.....	5
3.1.3 Software interfaces.....	5
3.1.4 Communications interfaces	5
3.2 Functional Requirements.....	5
3.2.1 User Class 1 – The Guest	5
3.2.2 User Class 2 – The Student.....	8
3.2.3 User Class 3 – The Establishment	13
3.2.4 User Class 4 – The Administrator	15
3.3 Non Functional Requirements	16
3.4 Performance Requirements.....	17

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 “Envision” 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 for the development team.

1.2 Scope

The “Envision” is a web-based application that can be accessed by anyone at anytime from anywhere. “Envision” helps students to find relevant internship or project based on the students search criteria and specifications like stipend, internship type, start date and more. It also provides a better approach for institutes to find reliable interns.

1.3 Definition, Acronyms and Abbreviations

Figure-1 Definition Table

TERM	DEFINITION
Database	Collection of information monitored by the system.
Field	A cell within a form.
User	Someone who interacts with the system. Can be a guest or a student or an establishment.
Student	A logged in user searching for an internship or a project or currently working in one.
Establishment	A logged in user which is an entity that already has or wants to post an advert for an internship.
Guest	A user who is yet not logged in.
Admin/Administrator	A user with specific permission for managing and controlling the system.
Web portal	A web application that provides specialized features.
Stake Holder	Anyone who benefits from the software.
Software Requirements Specification	A document that completely describes all the functions of a proposed system and the constraints under which it must operate. For example, this document.

DESC	Description
DEP	Dependency.
INP	Input
OUT	Output

1.4 References

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

1.5 Overview

The remainder of this document includes two chapters.

- The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.
- The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

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

The product is an entirely new product, it is not a part of a larger system. The product is supposed to be free of use for all, though the services are constrained on the user type and whether the user is logged-in or not. It is a web based system implementing client-server model.

2.2 Product architecture

This system will consist of two parts:

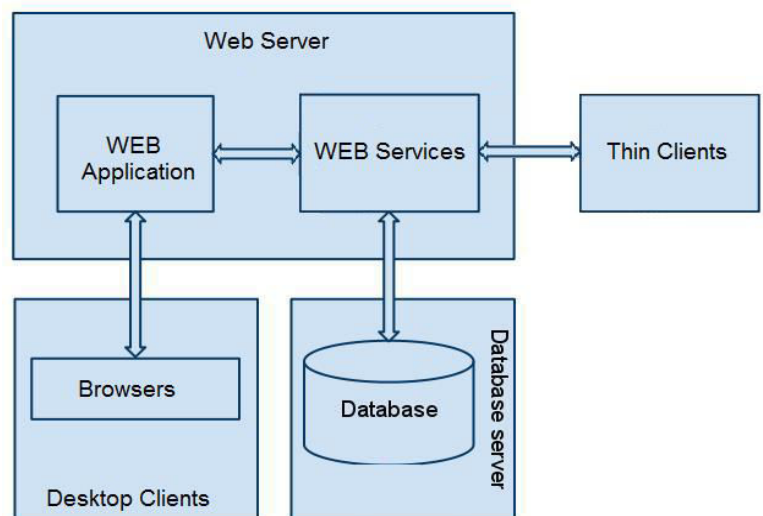
- Web application/portal
- A group of web services.

The web application will directly interact with the user and will be used to find different internships, projects or interns and view information about them depending on the user, while the web services will be used to provide and manage the information and the system.

The web services provide data to the web application, which the web application uses to display it to the user. As web services' work is to provide data and is

independent of view, these services could be used in the future by thin clients like mobile phones, tablets which could display this data according to their view. The functionality provided by the web services would be embedded into the web application in order for the user to be able to use the functions in the application in a seamless manner.

Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. The web services would directly communicate with the database.



2.3 Product functions

With the web application,

- Establishments will be able to find interns by posting an internship advert through a web-portal. This information will act as the bases for the search results displayed to the students.
- The students will be able to search for internships or projects. The result will be based on the criteria of user inputs. There are several search criteria and it will be possible for the administrator of the system to manage the options for those criteria that. The result of the search will be viewed in a list view. The list view will have one list item for each internship or project matching the search criteria and shows a small part of the posts' information so the user can identify the post.

- An administrator also uses the web-portal in order to administer the system and keep the information accurate. The administrator can, for instance, verify establishments and manage user information.

All system information is maintained in a database, which is located on a web-server. The software would facilitate communication between students and establishments via emails.

2.4 User characteristics

There are four types of users that interact with the system: guest, student, establishment and administrator. Each of these four types of users has different use of the system so each of them has their own requirements and specifications.

Guest can use the system to simply view different internships and projects but are not allowed to join any. They can register themselves or login into the system.

Students have all the perks of a guest user. This means that the user will have to be able to search for internships or projects, choose an internship or project from that search and then navigate to it.

On top of it a student can apply for an internship or join a project.

In order for the users to get a relevant search result there are multiple criteria that the users can specify and results would be displayed according to these criteria.

The Establishments will use the web application to manage the information about their internship or company, for example a description of the internship, contact information, short listing candidates for the internship and so on.

The administrators also only interact with the web application. They are managing the overall system so there is no incorrect information within it. The administrator can manage the information for each internship, projects as well as the options for both the students and establishment.

2.5 Constraints

The system is constrained to internet connection. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function. Both the web services and the web application will be constrained by the capacity of the database.

2.6 Assumptions and dependencies

One assumption about the product is that it will always be used on devices that have enough performance. If the device does not have enough hardware resources available for the application, for example the users might have allocated them to other applications, there may be scenarios where the application does not work as intended or even at all.

3. Specific requirements

This section contains all of 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

Each time the web application is requested the user should land on the home page i.e. the landing page of the application. Using this page the user would be able to choose various options like if the user is a student or a guest then searching for an internship or project, if the user is an establishment then managing the internships and applicants.

3.1.2 Hardware interfaces

Since neither the web application nor the web portal have any designated hardware, it does not have any direct hardware interfaces. The hardware connection to the database server is managed by the underlying operating system on the client device and the web server.

3.1.3 Software interfaces

The Web application uses several 3rd party APIs like Google Places API, Google Maps API for better interaction with the user.

3.1.4 Communications interfaces

The communication between the different parts of the system is important since they depend on each other. The communication between the database and the web services consists of operation concerning both reading and modifying the data which persists even with system failure, while the communication between the web services and the web application consists of simple reading and writing operations to pass information.

3.2 Functional Requirements

This section includes the requirements that specify all the fundamental actions of the software

3.2.1 User Class 1 – The Guest

Functional Requirement 1

Secure Login and Authentication

ID: FR 1.1
TITLE: Registration
INPUT: user_credentials
OUTPUT: status_flag, status_msg

DESC: If the user is not a registered user the system should allow registration of this new user. During registration the user has to enter the following details

- User Type (Student, Company)
- Company Name or User Name
- Email ID
- Password

ID: FR 1.2
TITLE: Log-in
INPUT: user_email, user_password
OUTPUT: error_flag, error_msg
DESC: The user should be able to log-in by providing email id and password. The system would automatically identify the user type and take actions according to it.

ID: FR 1.3
TITLE: Forgotten Password
INPUT: user_email
OUTPUT: status_msg
DESC: The system should provide mechanism for a user to enter the system if the user has forgotten his/her password.

Functional Requirement 2

View Internships

ID: FR 2.1
TITLE: Internship-Search
INPUT: keyword_list
OUTPUT: internship_list
DESC: The user should be able to search for internships through keywords

ID: FR 2.2
TITLE: Internship-Filtering
INPUT: filter_criteria
OUTPUT: internship_list
DESC: Given that the user is on the internship page, the user can filter out internships to narrow down their search. Filtering of internship could be done using the following criteria.

- Internship In (Start up, Established Company, NGO, Research Sector)
- Category of Internship (Graphic Designing, Marketing, Sales, Content Writing, Data Mining, etc.)
- Type of Internship (Full Time, Part Time, Work From Home)
- Start Date Between
- Duration of internship
- Perks of Internship (LOR, Certificate, Stipend, Job proposal, etc.)

ID: FR 2.3
TITLE: Internship-Result
INPUT: search_criteria
OUTPUT: internship_list
DESC: Given that the user is on the internship page the various internships should be displayed in a list view. Each internship list element would be a short description of the internship, it should have the following information.

- Internship post name
- Company providing the internship
- Starting date of the internship
- End date to apply for internship
- Minimum Duration of the internship
- Type of internship (Full time or Part Time)
- Locations of the internship
- Stipend
- Available seats

 Details button (To know details about the internship)

ID: FR 2.4
TITLE: Internship-details
INPUT: internship_id
OUTPUT: internship_info
DESC: Given that a user has selected a particular internship the following details about the internship is a must

- Company background
- What work would the user be doing in the internship
- Internship Prerequisites
- Internship Perks (LOR, certificate, Job proposal, etc.)

Functional Requirement 3

View Projects

ID: FR 3.1
TITLE: Project-Search
INPUT: keyword_list, category_list
OUTPUT: project_list
DESC: The user should be able to search for projects simply by keywords or category

ID: FR 3.2
TITLE: Project-Result
INPUT: search_criteria
OUTPUT: project_list

- DESC:** Given that the user is on the project page the various project should be displayed in a tabular view. Each project cell element would be a short gist of the project, it should have the following information.
- Project prototype Image
 - Project name
 - Project Category (Artificial Intelligence, Robotics, etc.)
 - Short description of the project
 - Current number of members in the project
 - Number of project likes.
 - Join now button (Disabled cause user not logged in)
- ID:** FR 3.3
- TITLE:** Project-details
- INPUT:** project_id
- OUTPUT:** project_info
- DESC:** Given that a user has selected a particular project the following details about the project is a must
- Project Summary
 - Projects final outcome
 - Project current main goal
 - Project current problems

3.2.2 User Class 2 – The Student

Functional Requirement 1

Manage a portfolio

Functional Requirement 1.1

Manage Education

- ID:** FR 1.1.1
- TITLE:** Add-Education
- INPUT:** education_details
- OUTPUT:** updated_portfolio
- DESC:** Student should be able to add an education detail

- ID:** FR 1.1.2
- TITLE:** Edit-Education
- INPUT:** education_id
- OUTPUT:** updated_portfolio
- DESC:** Student should be allowed to edit an education detail

- ID:** FR 1.1.3
- TITLE:** Delete-Education
- INPUT:** education_id

OUTPUT: updated_portfolio
DESC: The student should be able to delete an education

Functional Requirement 1.2

Manage Work Experience

ID: FR 1.2.1
TITLE: Add-WorkExperience
INPUT: work_experience_details
OUTPUT: updated_portfolio
DESC: Student should be able to add a work experience detail

ID: FR 1.2.2
TITLE: Edit-WorkExperience
INPUT: work_experience_id
OUTPUT: updated_portfolio
DESC: Student should be allowed to edit a work experience detail

ID: FR 1.2.3
TITLE: Delete-WorkExperience
INPUT: work_experience_id
OUTPUT: updated_portfolio
DESC: The student should be able to delete a work experience

Functional Requirement 1.3

Manage Volunteer Work

ID: FR 1.3.1
TITLE: Add-VolunteerWork
INPUT: volunteer_work_details
OUTPUT: updated_portfolio
DESC: Student should be able to add a volunteer work detail

ID: FR 1.3.2
TITLE: Edit- VolunteerWork
INPUT: volunteer_work_id
OUTPUT: updated_portfolio
DESC: Student should be allowed to edit a volunteer work detail

ID: FR 1.3.3
TITLE: Delete- VolunteerWork
INPUT: volunteer_work_id
OUTPUT: updated_portfolio
DESC: The student should be able to delete a volunteer work

Functional Requirement 1.4***Manage Certifications***

- ID:** FR 1.4.1
TITLE: Add-Certification
INPUT: certification_details
OUTPUT: updated_portfolio
DESC: Student should be able to add a certification detail
- ID:** FR 1.4.2
TITLE: Edit-Certification
INPUT: certification_id
OUTPUT: updated_portfolio
DESC: Student should be allowed to edit a certification detail
- ID:** FR 1.4.3
TITLE: Delete-Certification
INPUT: certification_id
OUTPUT: updated_portfolio
DESC: The student should be able to delete a certification

Functional Requirement 1.5***Manage Details***

- ID:** FR 1.5.1
TITLE: Add-Details
INPUT: user_details
OUTPUT: updated_portfolio
DESC: Student should be able to add certain details of his like mobile no, contact email, etc.
- ID:** FR 1.5.2
TITLE: Edit-Details
INPUT: user_id, user_details
OUTPUT: updated_portfolio
DESC: Student should be allowed to edit certain details of themselves
- ID:** FR 1.5.3
TITLE: Delete-Details
INPUT: user_id
OUTPUT: updated_portfolio
DESC: The student should be able to delete certain details like mobile no, contact email, etc.

Functional Requirement 2

View Internships

ID: FR 2.1
TITLE: Internship-Search
INPUT: keyword_list
OUTPUT: internship_list
DESC: The user should be able to search for internships through keywords

ID: FR 2.2
TITLE: Internship-Filtering
INPUT: filter_criteria
OUTPUT: internship_list
DESC: Given that the user is on the internship page, the user can filter out internships to narrow down their search. Filtering of internship could be done using the following criteria.

- Internship In (Start up, Established Company, NGO, Research Sector)
- Category of Internship (Graphic Designing, Marketing, Sales, Content Writing, Data Mining, etc.)
- Type of Internship (Full Time, Part Time, Work From Home)
- Start Date Between
- Duration of internship
- Perks of Internship (LOR, Certificate, Stipend, Job proposal, etc.)

ID: FR 2.3
TITLE: Internship-Result
INPUT: search_criteria
OUTPUT: internship_list
DESC: Given that the user is on the internship page the various internships should be displayed in a list view. Each internship list element would be a short description of the internship, it should have the following information.

- Internship post name
- Company providing the internship
- Starting date of the internship
- End date to apply for internship
- Minimum Duration of the internship
- Type of internship (Full time or Part Time)
- Locations of the internship
- Stipend
- Available seats

Details button (To know details about the internship)

ID: FR 2.4
TITLE: Internship-details
INPUT: internship_id
OUTPUT: internship_info

DESC: Given that a user has selected a particular internship the following details about the internship is a must

- Company background
- What work would the user be doing in the internship
- Internship Prerequisites
- Internship Perks (LOR, certificate, Job proposal, etc.)

ID: FR 2.5

TITLE: Apply-Internship

INPUT: internship_id

OUTPUT: approval_status

DESC: The student should be able to apply for an internship.

Functional Requirement 3

View Projects

ID: FR 3.1

TITLE: Project-Search

INPUT: keyword_list, category_list

OUTPUT: project_list

DESC: The user should be able to search for projects simply by keywords or category

ID: FR 3.2

TITLE: Project-Result

INPUT: search_criteria

OUTPUT: project_list

DESC: Given that the user is on the project page the various project should be displayed in a tabular view. Each project cell element would be a short gist of the project, it should have the following information.

- Project prototype Image
- Project name
- Project Category (Artificial Intelligence, Robotics, etc.)
- Short description of the project
- Current number of members in the project
- Number of project likes.
- Join now button (Disabled cause user not logged in)

ID: FR 3.3

TITLE: Project-details

INPUT: project_id

OUTPUT: project_info

- DESC:** Given that a user has selected a particular project the following details about the project is a must
- Project Summary
 - Projects final outcome
 - Project current main goal
 - Project current problems
- ID:** FR 3.4
- TITLE:** Create-project
- INPUT:** project_details
- OUTPUT:** project_list
- DESC:** The student should be able to start a project. The student would be duped as the owner/leader of the project. During creation of a project the following things would be asked to the leader
- Project title.
 - Project category (Robotics, Gaming, Designing, etc.).
 - Project Image.
 - A detailed description of the project,
 - The end product of the project.
 - The current goals of the project.
 - The current problems/requirements of the project.
- ID:** FR 3.5
- TITLE:** Edit-Project
- INPUT:** project_id
- OUTPUT:** project_info
- DESC:** The owner of the project should be able to make changes in the project like changing current goals and requirements.
- ID:** FR 3.6
- TITLE:** Join-Project
- INPUT:** project_id
- OUTPUT:** status_flag
- DESC:** The student should be able to join a project.

3.2.3 User Class 3 – The Establishment

Functional Requirement 1

Manage Company portfolio

- ID:** FR 1.1
- TITLE:** Edit-Company Details
- INPUT:** company_details
- OUTPUT:** updated_details

DESC: The system should provide mechanism for editing company details like the company logo or the company's background.

Functional Requirement 2

Manage Internship

ID: FR 2.1
TITLE: Add-Internship
INPUT: internship_details
OUTPUT: status_flag
DESC: The Establishment is allowed to add an advert of an internship. The following details about the internship during its creation would be asked to be filled up

- Internship Designation
- Internship Description
- Prerequisites for the internship
- Benefits by applying for that internship (LOR, Certification, etc.)
- Stipend
- Starting date of the internship
- Minimum duration of the internship
- Date before which a student must apply for an internship
- Total number of seats for internship

ID: FR 2.2
TITLE: Edit-Internship
INPUT: internship_id
OUTPUT: internship_info
DESC: The establishment should be able to edit an internship advert

ID: FR 2.3
TITLE: Delete-Internship
INPUT: internship_id
OUTPUT: status_flag
DESC: The establishment should be able to delete an internship

Functional Requirement 3

Manage Applicants

ID: FR 3.1
TITLE: View-Applicants
INPUT: internship_id
OUTPUT: applicant_list
DESC: The system should provide a mechanism for institutes to bifurcate various applicants for their internships.

ID: FR 3.2

TITLE: View-Applicants Profile
INPUT: applicant_id
OUTPUT: applicant_profile_info
DESC: The system should provide a mechanism for institutes to bifurcate various applicants for their internships.

ID: FR 3.3
TITLE: Approve Applicants
INPUT: applicant_id, approve_status
OUTPUT: updated_list
DESC: The system should provide a mechanism for institutes to bifurcate various applicants for their internships.

3.2.4 User Class 4 – The Administrator

ID: FR 1
TITLE: Verify Company
INPUT: company_details
OUTPUT: verified_status
DESC: In order for a company to post an internship advert, the administrator must verify the company first.

Functional Requirement 2

Manage Company Types List

ID: FR 2.1
TITLE Add Type
INPUT: type_name
OUTPUT: type_list
DESC: The administrator must be able to add a new type of company.

ID: FR 2.2
TITLE Edit Type
INPUT: type_id, type_name
OUTPUT: type_list
DESC: The administrator must be edit a company type if it is wrongly written.

ID: FR 2.3
TITLE Delete Type
INPUT: type_id
OUTPUT: type_list
DESC: The administrator must be able to delete a company type.

Functional Requirement 3*Manage Categories List for internships and projects*

ID:	FR 3.1
TITLE	Add Category
INPUT:	category_name
OUTPUT:	category_list
DESC:	The administrator must be able to add a new category
ID:	FR 3.2
TITLE	Edit Category
INPUT:	category_id, category_name
OUTPUT:	category_list
DESC:	The administrator must be edit acategory if it is wrongly written.
ID:	FR 3.3
TITLE	Delete Category
INPUT:	category_id
OUTPUT:	category_list
DESC:	The administrator must be able to delete category.

Functional Requirement 4*Manage Users*

ID:	FR 4.1
TITLE	Delete User
INPUT:	user_id
OUTPUT:	status_flag
DESC:	The administrator must be able to delete a user.
ID:	FR 4.2
TITLE	Edit User Role
INPUT:	user_id, user_role
OUTPUT:	status_flag
DESC:	The administrator should be able to change a user's role.

3.3 Non Functional Requirements

The System will be on a server with high speed Internet capability. The software developed here assumes the use of a tool such as IIS for connection between the Web pages and the database. The speed of the user's connection will depend on the hardware used rather than characteristics of this system.

3.4 Performance Requirements

The requirements mentioned in this section provide a detailed overview of the restrictions placed on the performance of the system.

ID: PR 1

TITLE: Prominent-Search

DESC: The search bar should be easy to access by the user.

ID: PR 2

TITLE: Filter-bar

DESC: The filtering criteria should be easy to understand and use by the user.

ID: PR 3

TITLE: System Dependability

DESC: If the system losses access to the internet or gets some faulty input then the user must appropriately be informed.