The Software Design Specification Outline

# Introduction

## 1.1 Purpose

The purpose of this document is to describe the design of the system in complete detail so that the software can be built with a mutual understanding and each function to be used is explained. The website is aimed at connecting entrepreneurs, start-ups and interns in the university, and at the same time providing students with the knowledge of placements, interview questions and recruitment criteria of various companies so that students could be better prepared and equipped at job interviews.

## 1.2 Scope

This document describes functional details of the website,  focusing on how each function should be implemented and the base level of the system.

## 1.3 Definition, Acronym and Abbreviations

Cohesion - Cohesion refers to the degree to which the elements of a module belong together. Thus, cohesion measures the strength of relationship between pieces of functionality within a given module.

Coupling - Coupling is the degree of interdependence between software modules, a measure of how closely connected two routines or modules are; the strength of the relationships between modules.

Profanity Filter - A swear filter, also known as a profanity filter or language filter is a software subsystem which modifies text to remove words deemed offensive by the administrator or community of an online forum.

Session -A login session is the period of activity between a user logging in  and loggingout of a (multi-user) system.

Required Field - A field in the data set of a document which is required for successful document generation.

Sequence Diagram - A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It shows object interactions arranged in time sequence.

Class Diagram - A class diagram is a static diagram and represents the static view of an application. It is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML).

State Diagram - State diagram is used to describe the behaviour of a single object in response to series of events in the system.

## 1.4 References

1. For API References in Django(For Backend) -

https://docs.djangoproject.com/en/1.10/ref/

## 1.5 Overview of Document

The software design document is divided into  ---- sections with various sub-sections:

i. Index

1. Introduction - It highlights the purpose and scope of the document, also explains the acronyms that are used throughout the document.

2. System Architecture Description - It describes the major architectural decisions that are taken regarding the project , it explains various modules and functions that make the webpage. It also includes the Sequence Diagrams, Class Diagrams and State Diagrams along with User-Interface details.

3. Description of Components - Every component in the project is explained here in detail.

4. Reuse and Relationships to other Products - It lists the modules and functions that will be shared by multiple processes.

5. Design Decisions - It includes various designs being used in the project.

6. Pseudocode for Components

7. Appendix

# 2. System architecture description:

For agility while the design is in progress and to address the issues almost as soon as they arise, the entire system has been strategically divided into sub systems (modules) .

There are three major modules,

1) The Placements Platform,

2) The StartUps Platform and

3) INternships Platform.

A higher degree of cohesion has been mantained at an inter module level , since apart from 2 cases at max, there are strictly no interdependencies among the modules of the software.

In correspondance with the aforesaid point , we have also tried to mantain a lowest level of coupling possible.

Hierarchicaly speaking , the number of levels defined for classes is not more than 4 .

In an attempt to introduce optimization , we have compromised cohesion , such that there shall be single database for profiles , that shall be shared among the 3 different modules. This stack like approach not only streamlines the user’s approach to the system but also greatly reduces a maintenance engineer’s work. It is evident from earlier structures like IMDB , how an array , with pointers to handle permissions of approach by user or an external system, can indeed end up as excillent optimisation solutions.

**Overview of modules:**

There are three main modules as listed:

* Internships platform
* Placements platform
* Startups platform

Internships platform:

Aims to aid students looking for internships.A student can log in , create , maintain a profile, with all his necessary details, that shall be viewable by those willing to offer internship.

A student may any time choose to discontinue his profile , and can get rid of it in something as easy as one click.

Entire system shall be equipped with a robust messaging system, for direct on / group communications.

Students going for same internships shall have the ability to form profile groups, which shall then unlock for them a forum for an all group discussion. Here ideas can be openly exchanged , subject to they being in proper way of conduct .

Both the messaging system and the forum shall be equipped with the profanity filter to ensure proper language is used. Moreover the system administrator shall have the freedom and discretion to kick a member out for inappropriate behaviour.

# Placements platform:

This module allows the user to search from among the companies that offer campus placements at the university. Once a company has been chosen the user shall now have the freedom to decide what he wants to know about the company. From company information , company profiles, to mock test papers and tips and tricks to crack through the interviews, you name it , this module has it.

It is meant to serve as a one stop shop for all students having getting placement into a good company in mind.

# Startups Platform:

This perhaps , no doubt is the most ambitious module of the entire software project. This module basically relies on profile data base and forum architecture of the first module to create a communication between students owning startups , or willing to start one of their own.

A student upon logging in shall be greeted by a dedicated tab meant for students trying to start new ventures , and need partners/co founders.

Apart from this , the hiring system , the messaging system , the forum , discussions and everything shall be a part of this system and shall be incorporated as it is.

Just provided the fact that the forum here shall be larger than one in the initiating module. This has purposely been done to allow allow even non participants, free thinkers, professors, tutors , mentors - all to have a common forum for discussions regarding sharpening of skills and perceptions.

# Structure and relationships:

1. Internships platform

From a user’s point of view:

What happens when a user logs into the system ? the following sequence chart aims to answer this question :

The vairious state choices are listed.

Additional tutorial help has been provided using an in site navigator.

Home Page

Internships Platform

Update profile info

Available internship

Update forum post

send/receive messages

From an administrator’s point of view:

An important revision includes incorporation of a technical help tab for the administrator, in order to obtain remote assistance.

Login

Maintain database

Moderate forums

Upkeep matchmaking tool

**2.) Placements platform:**

From a user’s point of view:

What happens when the user is not interested in internships ? but in placement ?

Following sequence diagram shows how our placement platform works (state to state).

Home screen

Select company

Eligibility criteria

Cracking interview

Cracking written

Previous year recruitment tech

**3.) StartUps Platform:**

What happens when a user logs on to the startups platform ??

The following sequence diagram attempts to answer this in a more of diagrammatic way.

The person receives choices depending upon his profile. Whether it is that of an intern, a guest or an entrepreneur.

The person receives corresponding options , which again range from people trying to conceive a start up , to a student already owning a startup.

Among those who already own a startup , there is going to be a database division among those who are looking for a co founder ( they already have an idea) , those who are looking for an investor (any one interested in funding the operations legally) and those who are looking for students to work as intern for them.

Moreover , forums here need to be robust enough to be accessible by all , and moreover , must be monitored as well, that is the reason why this guest profile system has been created here for the worth of it.

Home screen

Entrepreneur’s

platform

Login to profile

Owning startup

Looking for co.

Looking for interns

forums /discussion

messagin

## User Interface Issues:

Main principles behind this software’s user interface!

* User Interface: regular browser used to access moodle.
* Home screen: The first screen a user sees when logging onto the system
* Login screen: Immediately after having logged on , a user is presented with a screen that prompts him to pick up an alias (profile) even if it is a guest profile.
* Look for internships: a dedicated tab visible to a student logged in with an intern profile, allows the user to search for internships.
* Confirm internship: another dedicated key , which upon activation , sends a message to the admin , that certain guy is interested in certain internship, and so a communication must be established.
* In upcoming versions of software, this application shall be controlled by AI entirely.
* Update profile: any new addition to a student’s current status can easily be updated with the help of this yet another dedicated key.
* Updating of modules like profile picture, profile name etc: These are subsystems of the updating and validation system.
* View company information: performance, style of hiring ,type of people required , to cgpa bar , covers every aspect of information.
* mock test: timed test for testing upon questions (similars) generally put up by the company.
* Interview helpkit: appropriate behaviour to question guide for cracking interviews.
* The Forum : all sorts of discussions form all sorts of profiles.
* Messaging service: allows all users with appropreate permissions across the system to talk, discuss and exchange ideas.
* Delete profile : a user willing to discontinue the usage of this system may anytime completely delete his profile , with the tap of a single key.

# 3. Detailed description of components

3.1 Component Template Description

|  |  |
| --- | --- |
| Identification | The unique name for the component and the location of the component in the system. |
| Type | A module, a subprogram, a form, a data file, a control procedure, a class, etc |
| Purpose | Function and performance requirements implemented by the design component, including derived requirements. Derived requirements are not explicitly stated in the SRS - but are implied or adjunct to formally stated SDS requirements. |
| Subordinates | The internal structure of the component, the constituents of the component, and the functional requirements satisfied by each part. |
| Dependencies | How the component’s function and performance relate to other components. How this component is used by other components. The other components that use this component. Interaction details such as timing, interaction conditions (such as order of execution and data sharing), and responsibility for creation, duplication, use, storage, and elimination of components. |
| Interfaces | Detailed description of all external or internal interfaces as well as of any mechanism for communicating through messages, parameters, or common data areas. All error messages and error codes should be identified. All screen formats, interactive messages, and other user interface components (originally defined in the SRS) should be given here. |
| Resources | A complete description of all resources (hardware or software) external to the component but required to carry out its functions. |
| Processing | A full description of the functions presented in the Function subsection. Pseudocode can be used to document algorithms, equations, and logic. |
| Data | For the data internal to the component, describes the representation method, initial values, use, semantics, and format. |

3.2 Description of Login Screen

|  |  |
| --- | --- |
| Identification | LoginScreen |
| Type | Class/Form |
| Purpose | The login screen assures that only students , teachers and company officials can access the system. |
| Subordinates | This screen contains links to the following screens:   * Main Menu Screen * New User Account Screen |
| Dependencies | The following screen links to this screen:   * Main Menu Screen |
| Interfaces | The links are contained in the bottom half of the screen. The screen is designed to be easy to view using the resolution standard on the PDA. |
| Resources | Database Access Requirements: access to the violator information found in the appropriate database tables. Please see Appendix A for a description of the information is associated with a violator. |
| Processing | The only type of processing required is inputting information into the text boxes and navigating to other forms using links in the bottom half of the screen. Each link directs the user to a different screen that corresponds to the link that the user selects. |
| Data | The data for the Login Screen is the username and password entered by the user. It is validated with a query against the database. |

3.3 Description of New User Account Screen

|  |  |
| --- | --- |
| Identification | NewUserAccountScreen |
| Type | Class/Form |
| Purpose | The new user account screen allows students , teachers and company officials the ability to create a unique user name for themselves, which can be used to log into the system. |
| Subordinates | This screen contains links to the following screen:   * Login Screen |
| Dependencies | The following screen links to this screen:   * Login Screen |
| Interfaces | The links are contained in the bottom half of the screen. The screen is designed to be easy to view using the resolution standard on the PDA. |
| Resources | Database Access Requirements: access to the vehicle registration information found in the appropriate database tables. This access is used to create a new user and check to make sure not the same as another user. |
| Processing | The only type of processing required is inputting information into the text boxes and navigating to other forms using links in the bottom half of the screen. Each link directs the user to a different screen that corresponds to the link that the user selects. |
| Data | The data supplied by the system are fields the new user must enter. The data given by the user is the appropriate information needed to fill in the given fields. This data once determined valid, by checking to make sure the user does not already exist, is saved in the database. |

3.4 Description of Main Menu Screen

|  |  |
| --- | --- |
| Identification | MainMenuScreen |
| Type | Class/Form |
| Purpose | The main menu screen assists the user by presenting them with tasks the program performs. |
| Subordinates | This screen contains links to the following screens:   * Profile * Information |
| Dependencies | The following screens link to this screen:   * Login Screen * Info screen * Profile screen |
| Interfaces | The links are contained in the bottom half of the screen. The screen is designed to be easy to view using the resolution standard on the PDA. |
| Resources | None |
| Processing | The only type of processing required is inputting information into the text boxes and navigating to other forms using links in the bottom half of the screen. Each link directs the user to a different screen that corresponds to the link that the user selects. |
| Data | There is no data entered for this screen. |

3.5 Description of LookupStartupInfoScreen

|  |  |
| --- | --- |
| Identification | LookupStartupInfoScreen |
| Type | Class/Form |
| Purpose | The lookup vehicle information screen allows the user to enter a username or a startup name to retrieve student/faculty/startup profile and other deatails. |
| Subordinates | This screen contains links to the following screens:   * Main Menu Screen * Profile Screen |
| Dependencies | The following screen links to this screen:   * Main Menu Screen |
| Interfaces | The links are contained in the bottom half of the screen. The screen is designed to be easy to view using the resolution standard on the PDA. |
| Resources | Database Access Requirements: access to the violator information found in the appropriate database tables. |
| Processing | The only type of processing required is inputting information into the text boxes and navigating to other forms using links in the bottom half of the screen. Each link directs the user to a different screen that corresponds to the link that the user selects. |
| Data | The data entered by the user for this screen is the vehicle in question’s license plate and state. |

3.6 Description of InterviewInfo

|  |  |
| --- | --- |
| Identification | InterviewInfo |
| Type | Class/Form |
| Purpose | The InterviewInfo allows the user to lookup various interview related information |
| Subordinates | This screen contains links to the following screens:   * Main Menu Screen * Profile * Home |
| Dependencies | The following screens link to this screen:   * Main Menu Screen * Profile |
| Interfaces | The links are contained in the bottom half of the screen. The screen is designed to be easy to view using the resolution standard on the PDA. |
| Resources | Database Access Requirements: access to the violator information and ticket information appropriate database tables.. |
| Processing | The only type of processing required is inputting information into the text boxes and navigating to other forms using links in the bottom half of the screen. Each link directs the user to a different screen that corresponds to the link that the user selects. |
| Data | The data related to Interview info stored in the database |

# 4. Reuse and relationships to other products

The software is built with the definition of following frameworks:

1) The database is designed using the SQLite framework.

2) The languages (html, CSS, java script, Python) are built upon the Visual Studio 2015 Enterprise software.

3) The software is majorly designed upon the Microsoft Windows OS. However, it is created as a cross-platform forum to work over Linux and iOS as well.

4) Protocols that interface between the front-end and back-end codes. Any platform supporting the above features is compatible with the designated app.

# 5. Design decisions and tradeoffs

We have used HTML, CSS ,Bootstrap and JS for designing the website.

At backend we used the Python based Django Framework , using SQLITE for the database.

We are using Elasticsearch for unstructured search.

Elasticsearch is a distributed, RESTful search and analytics engine capable of solving a growing number of use cases

# 6. Pseudo code for components

# 7. Appendices (if any)