Software Requirements Document (SRS)

Project Title

Abacus Academy Admission Portal

Project ID - B1JA12

Team Name – Avengers

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver No.** | **Change Description** | **Changed Sections** | **Author** |
| 1.00 | Initial version | NA |  |
|  |  |  |  |

Team Structure

|  |  |  |  |
| --- | --- | --- | --- |
| **Employee Name** | **Employee ID** | **Email ID** | **Mobile** |
| Gaurav Kumar Singh | 8118225 | gauravkumarsingh@virtusa.com | 9931496286 |
| Varalakshmi Trilokya Konduri | 8117867 | varalakshmitrilokya@virtusa.com | 8106257966 |
| Aman Thakur | 8118282 | amant@virtusa.com | 8292104146 |
| Kirthik Roshan R | 8118426 | kirthikroshanr@virtusa.com | 9750481619 |
| Dinesh M | 8117265 | dineshms@virtusa.com | 6382517748 |
| Megha Raj Yadav | 8118601 | meghayadav@virtusa.com | 6294584104 |
| Sharath Chandra Bandela | 8117465 | sharathbandela@virtusa.com | 8125872710 |
| A.Sri Shiva Ganga | 8117872 | srishivaganga@virtusa.com | 9705765595 |

Overview of the Project

Our project aim is to create a Full Stack Application using Angular namely " Abacus Academy Admission Portal " where a user can register and get itself enrolled in any course(s) from the course list and selecting any institution from the list in the system after approval of the submission by the admin .User can also give their review/feedback in the portal.

Actors of the System

|  |  |
| --- | --- |
| **Actor** | **Responsibilities** |
| Admin | Admin is a human who interacts with the system and manages everything like-adding, editing, deleting of courses ,students and academies. |
| User | User is a human who interacts with the system using the interface provided, and majorly takes admission in various courses listed in the system that’s going to be the key business driver of the entire system. |

Functional Requirements

|  |  |
| --- | --- |
| **Use Case #1** | User registers in the system |
| **Actor** | User |
| **Successful Flow** | 1. The user lands on the home page of the system.  2. The user views the option either to login or register in the system.  3. The user selects to register.  4. The system displays option to enter the username, email id, mobile no and password.  5. Upon submission, the system displays a registration success message , along with customer details entered in the previous step |
| **Exceptional Flow** | 1.When invalid characters are entered in user name, email id , mobile  And password line fields, system throws the relevant error message.  2. If the registration process is not successful for any reason, system informs the user with relevant error message |
| **Outcome** | A user account is created in the system |

|  |  |
| --- | --- |
| **Use Case #2** | User logs in, in the system |
| **Actor** | User |
| **Successful Flow** | 1. The user lands on the login page of the system.  2. The user views the option either to login or register in the system.  3. The user selects to login.  4. The system displays option to select the role (user/admin ,select role user) and enter the email id and password.  5. Upon submission, user lands on the home page of the system. |
| **Exceptional Flow** | 1. When invalid characters are entered in email id and password line fields, system throws the relevant error message.  2. If the login process is not successful for any reason, system informs the user with relevant error message. |
| **Outcome** | A user login in the system. |

|  |  |
| --- | --- |
| **Use Case #3** | User adds admission. |
| **Actor** | User |
| **Successful Flow** | 1. The user lands on the home page of the system.  2. The user clicks the button academy.  3. The user selects an academy.  4. The user searches for a course.  5. The user selects the course.  6. The user fills the admission form.  7. Upon submission, user successfully registers in the course. |
| **Exceptional Flow** | 1. When invalid characters are entered in the admission form, system throws the relevant error message. |
| **Outcome** | The user successfully registers for the admission. |

|  |  |
| --- | --- |
| **Use Case #4** | User views admission. |
| **Actor** | User |
| **Successful Flow** | 1. The user lands on the home page of the system.  2. The user clicks on the tab enrolled course.  3. The user views the list of enrolled courses.  4. The user clicks on any course.  5. The admission details regarding that course is retrieved. |
| **Exceptional Flow** | N.A |
| **Outcome** | The user successfully views the admission details for a particular course. |

|  |  |
| --- | --- |
| **Use Case #5** | User edits admission. |
| **Actor** | User |
| **Successful Flow** | 1. The user lands on the home page of the system.  2. The user clicks on the tab enrolled course.  3. The user views the list of enrolled courses.  4. The user clicks on any course.  5. The admission details regarding that course is retrieved.  6. The user clicks on edit button.  7. The user edits the admission and submits. |
| **Exceptional Flow** | 1. When invalid characters are entered in the admission form, system throws the relevant error message.    2. If the admission status is approved then user can’t edit the admission (user can only view). |
| **Outcome** | The user successfully edits the admission details for a particular course. |

|  |  |
| --- | --- |
| **Use Case #6** | User deletes admission. |
| **Actor** | User |
| **Successful Flow** | 1. The user lands on the home page of the system.  2. The user clicks on the tab enrolled course.  3. The user views the list of enrolled courses.  4. The user clicks on delete button.  5. Admission details for the particular course gets deleted. |
| **Exceptional Flow** | N.A |
| **Outcome** | The user successfully deletes the admission details for a particular course. |

|  |  |
| --- | --- |
| **Use Case #7** | User views admission status. |
| **Actor** | User |
| **Successful Flow** | 1. The user lands on the home page of the system.  2. The user clicks on the tab enrolled course.  3. The user views the list of enrolled courses and admission status of the particular course. |
| **Exceptional Flow** | N.A |
| **Outcome** | The user successfully views the admission status for a particular course. |

|  |  |
| --- | --- |
| **Use Case #8** | User gives review. |
| **Actor** | User |
| **Successful Flow** | 1. The user lands on the home page of the system.  2. The user clicks on the tab review.  3. The user submits the form by providing ratings and reviews. |
| **Exceptional Flow** | 1. When invalid characters are entered in the admission form, system throws the relevant error message. |
| **Outcome** | The user successfully reviews the portal. |

|  |  |
| --- | --- |
| **Use Case #9** | Admin logs in, in the system |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the login page of the system.  2. The admin views the option either to login or register in the system.  3. The admin selects to login.  4. The system displays option to select the role (user/admin ,select role admin) and enter the email id and password.  5. Upon submission, admin lands on the home page of the system. |
| **Exceptional Flow** | 1. When invalid characters are entered in email id and password line fields, system throws the relevant error message.  2. If the login process is not successful for any reason, system informs the user with relevant error message. |
| **Outcome** | The admin logs in, in the system. |

|  |  |
| --- | --- |
| **Use Case #10** | Admin adds course. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the course tab.  3. The list of available courses are displayed.  4. Admin clicks on add course button.  5. Admin fills the form for course details and clicks add course.  5. Upon submission, the course gets added on the course list. |
| **Exceptional Flow** | 1. When invalid characters are entered in the form, system throws the relevant error message. |
| **Outcome** | The admin successfully adds the new course in the system. |

|  |  |
| --- | --- |
| **Use Case #11** | Admin views courses. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the course tab.  3. The list of available courses are displayed. |
| **Exceptional Flow** | N.A |
| **Outcome** | The admin successfully views the available courses in the system. |

|  |  |
| --- | --- |
| **Use Case #12** | Admin edits course. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the course tab.  3. The list of available courses are displayed.  4. Admin clicks on edit button on the particular course.  5. Admin fills the form for course details and clicks edit course.  6. Upon submission , the course details gets edited on the course list. |
| **Exceptional Flow** | N.A |
| **Outcome** | The admin successfully edits the course details in the system. |

|  |  |
| --- | --- |
| **Use Case #13** | Admin deletes course. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the course tab.  3. The list of available courses are displayed.  4. Admin clicks on delete button on the particular course.  5. The course details gets deleted from the course list. |
| **Exceptional Flow** | N.A |
| **Outcome** | The admin successfully deletes the course details from the system. |

|  |  |
| --- | --- |
| **Use Case #14** | Admin adds academy. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the academy tab.  3. The list of available academies are displayed.  4. Admin clicks on add academy button.  5. Admin fills the form for academy details and clicks add academy.  5. Upon submission , the academy gets added on the academy list. |
| **Exceptional Flow** | N.A |
| **Outcome** | The admin successfully adds the new academy details in the system. |

|  |  |
| --- | --- |
| **Use Case #15** | Admin views academies. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the academy tab.  3. The list of available academies are displayed. |
| **Exceptional Flow** | N.A |
| **Outcome** | The admin successfully views the available academies in the system. |

|  |  |
| --- | --- |
| **Use Case #16** | Admin edits academy. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the academy tab.  3. The list of available academies are displayed.  4. Admin clicks on edit button on the particular academy.  5. Admin fills the form for academy details and clicks edit course.  6. Upon submission, the academy details gets edited on the academy list. |
| **Exceptional Flow** | N.A |
| **Outcome** | The admin successfully edits the academy details in the system. |

|  |  |
| --- | --- |
| **Use Case #17** | Admin deletes academy. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the course tab.  3. The list of available academies are displayed.  4. Admin clicks on delete button on the particular academy.  5. The academy details gets deleted from the academy list. |
| **Exceptional Flow** | N.A |
| **Outcome** | The admin successfully deletes the academy details from the system. |

|  |  |
| --- | --- |
| **Use Case #18** | Admin adds student. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the student tab.  3. The list of all enrolled students are displayed.  4. Admin clicks on add student button.  5. Admin gets a list of all pending student enrolment.  6. Admin clicks on approve button.  5. The student gets added in the student list with a unique student id assigned. |
| **Exceptional Flow** | 1. Admin clicks on deny button and the students enrolment gets declined and not being added in the student list. |
| **Outcome** | The admin successfully add the student in the system. |

|  |  |
| --- | --- |
| **Use Case #19** | Admin views student. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the student tab.  3. The list of all enrolled students are displayed. |
| **Exceptional Flow** | N.A |
| **Outcome** | The admin successfully views the students list in the system. |

|  |  |
| --- | --- |
| **Use Case #20** | Admin edit student. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the student tab.  3. The list of all enrolled students are displayed.  4. Admin clicks on edit icon in a particular student row.  5. Admin edits the prefilled student details form.  6. Admin clicks on update student button.  5. The student details gets edited in the student list. |
| **Exceptional Flow** | 1. When invalid characters are entered in the form, system throws the relevant error message. |
| **Outcome** | The admin successfully edits the student details in the system. |

|  |  |
| --- | --- |
| **Use Case #21** | Admin delete student. |
| **Actor** | Admin |
| **Successful Flow** | 1. The admin lands on the home page of the system.  2. The admin clicks on the student tab.  3. The list of all enrolled students are displayed.  4. Admin clicks on delete icon in a particular student row.  5. The student details gets deleted from the student list |
| **Exceptional Flow** | N.A |
| **Outcome** | The admin successfully deletes the student details from the system. |

Software Tools

|  |  |  |  |
| --- | --- | --- | --- |
| **Tools/**  **Technologies Name** | **Version** | **Purpose** | **Remarks** |
| Angular | 10.0 | Frontend/UI development | https://angular.io/start |
| Spring Boot |  | Backend Development | https://spring.io/projects/spring-boot |
| Oracle |  | Database | https://www.oracle.com/in/index.html |