|  |
| --- |
| **CHANGE HISTORY** |

|  |  |  |
| --- | --- | --- |
| **DATE** | **SECTIONS CHANGED** | **CHANGE DESCRIPTION** |
| 2-5-15 | All | Initial Creation of SRS |
| 10-5-15 | Problem Statement, Functional Requirement | Added more points with descriptions |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **INTRODUCTION** |

|  |  |
| --- | --- |
| **Problem Statement**  **(Broad scope of the System)** | Create a Student Portal System for student to avail services provided by the enrolled institution.   * Information Inquiry: Display available trains between source & destination on a date range.   + Also,display Train timings   + Route Info   + Ticket prices in different classes   + Availability based on date & no. of passengers * Ticket booking based on availability:   + Book ticket on selected train   + In selected classes   + Between different places   + Record traveller details (Name, Age, Sex etc.)   + Infants (age < 2) are free   + Make Payment through Credit or Debit card * Amend Tickets   + Change travel date & class   + Add / Delete travellers * Cancel Ticket * Print Itinerary (for multiple trains)   + Print Ticket   + Print Train details |
| **Goals of Implementation**  **(Benefits of the system)** | * Manage and update user profile * Update existing to-do information * Seek career guidance provided by our company experts * Budget planning using calendar, payment getaways and e-wallets * Analysis of student reports, online platform assessment reports * Peek into the recent news feed, academic activities of the institution * Manage and maintain daily life schedule * Interact with peers using the Chat section |
| **Environmental Requirements** |  |
| **Hardware** | Intel i7, i5, 8 GB DDR3, 256 GB HDD |
| **Software** | React, MongoDB, Django, JAVA, |
| **OS/Platforms** | Windows 10, Android 12, MAC OS, IOS 15.1 |
| **Constraints**  **(Duration, Cost ... )** | 3 months, 6-member team, 1.2 lacs Rupees |

|  |
| --- |
| **FUNCTIONAL REQUIREMENTS** |

|  |  |
| --- | --- |
| **MODULE / MAJOR FEATURES #1**  **LOGIN /REGISTRATION & MAIN SCREENS** | * Display LOGIN/REGISTRATION screen with below details   + Ask if New **User** or Existing User   + If New User ask him/her to Register in Registration screen     - Collect Name, Address, Contact No., User Id & Password & Save in the system   + If Existing User ask him/her to Login through Login screen     - Collect User Id and Password & authenticate * After password validation,   + Allow user to enter the system & display the Main screen containing below info     - Menu options for all modules/major features     - Logo of the system |
| **MODULE / MAJOR FEATURES #2**  **PROFILE SECTION** | * When User selects the “Profile” option,   + He /She should be able to look at his/her information:     - Name     - Academic Institution Name     - Budget Planning     - Study Report Analysis   + Below Details For budget planning section will be displayed     - Dues     - Payment gateway     - Record of money shared between peers,   + Below details for Study report Analysis     - Payment gateway     - Result     - Attendance     - Online Rank |
| **MODULE / MAJOR FEATURES #3**  **TODO SECTION** | * When User selects the to-do section, he/she should be able to plan his/her tasks. * The user should be able to check on his/her upcoming events. |
| **MODULE / MAJOR FEATURES #4**  **CAREER GUIDANCE SECTION** | * When User selects the career guidance section, he/she should be able to look for internship opportunities. * The user should be able to get information about placement related queries. |
| **MODULE / MAJOR FEATURES #5**  **CHAT SECTION** | * When User selects the chat section, he/she should be able to send and receive messages from its peers . |
|  |  |
|  |  |

|  |
| --- |
| **NON-FUNCTIONAL REQUIREMENTS** |

|  |  |
| --- | --- |
| **USER INTERFACE RELATED** | Web-based Interface, Native application-based interface |
| **PERFORMANCE RELATED** | Response Time limit for Inquiries: 0.5 secs |
| **REUSABILITY RELATED** | News feed, Chat section have reusable functions |
| **RELIABILITY RELATED** | Stripe Payment, Chat Section has high reliability |
| **MAINTAINABILITY**  **RELATED** | System should be highly maintainable: Good technical documents, Proper commenting inside programs, Coding standards & guidelines must be followed |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| **CONCLUSION** |

|  |  |
| --- | --- |
|  |  |