

Software Design Specification

FOR

Share Karo App

Version 1.0 approved

Prepared by Share Karo App members:

Akshay Gupta: U101114FCS039

Anshul Sharma: U101114FCS046 (S2)

Ayushi Jain: U101114FCS055

Ishitaa Sayal: U101114FCS070 (S3)

Jatin Kakkar: U101114FCS072 (S3)

Jharana Shrivastava: U101114FCS073 (S3)

<NIIT UNIVERSITY>

<28 October 2016>

1. Introduction

1.1 Purpose of this document

The major focus of this SDS document is to establish an elaborated explanation of Share Karo App. The document will explain the purpose and key features of the app, the app interfaces at the view level, its functions, key constraints under which the app needs to operate and how the system tends to react to the external factors. This SDS document is targeted for the users, stakeholders and the developers of the app.

The document focuses upon the following:

- Class hierarchies and interactions
- Data flow and design
- Processing narratives
- Design constraints and restrictions
- User interface design
- Test cases and expected results

The document is intended for various readers such as developers, testers, project managers and documentation writers. App users are target audience and end viewers for the entire documentation and product.

1.2 Scope of the development project

Share Karo App is specifically designed for university students to provide them with an eloquent platform to share their means of leisure in hostel with other students. Its helps the students to know and interact with each other and exchange shareable items like novels, music, TV Series and study materials. It helps the student to socialize with people beyond their circle. The app is a generic sharing and socializing portal for local area where physical connection is easily established. The app majorly focuses on university students.

1.3 Definitions, acronyms, and abbreviations

S.no	Subject	Description
1.	Admin	The user which opens up the app for the locality and dispatches the login credentials, handling the access rights.
2.	Uploader (user)	The person who uploads data for others to share and download.

3.	Downloader (user)	The person who goes online and searches for data to download.
4.	Login Credentials	The login details that lets the user enter and use the app.
5.	Profile	The interests and descriptions about a person that defines the person's criteria.
6.	Routine	The schedule that one updates so as to achieve a travel companion.
7.	File	The files that the up loader uploads for others to view and share.
8.	Database	The back-end database that saves and retrieves data for people.
9.	Contact	The contact details of users that get displayed so as to collaborate personal meetings.

1.4 References

This System Requirements Specification document majorly refers to IEEE standards of documentation.

1.5 Overview of document

The document SDS is developed for the people who have been technically associated with the project such as software developers, team managers etc. The document defines the software functions. The functions that product is capable of performing are:

The user can sign up with the university/domain login and share their contact (email/phone no) if they feel like. The user needs to explain their interests in particular movie/novel genres so that other users can request them. The user is also allowed to upload the ppts and presentations they created for the juniors to refer. He/she can draw out the details of their journey so as to check if anybody else matches the same. The user can look up under specific

headings (entertainment, studies, conveyance) and check if their requirements match any of the ones uploaded and can then contact the concerned person. The user is also allowed to make requests to a specified person for contact regarding the requirement.

The overview of the document helps you to understand any part that specifically interests you.

PART 1	INTRODUCTION	This part provides details about the project's purpose and how it is designed.
PART 2	SYSTEM ARCHITECTURE DESCRIPTION	This section will provide a detailed account of all the modules that have been used in developing this software. In the end the user would get an idea of how would actually our software look like and how would it function.
PART 3	DETAILED DESCRIPTION OF COMPONENTS	In this part of the SDS a detailed description of all the classes, their attributes and the functions related to them have been described. What is the function of each and every function has been described in detail.
PART 4	REUSE AND RELATIONSHIP TO OTHER PRODUCTS	The way this app interacts with contemporary products.
PART 5	DESIGN DECISIONS AND TRADEOFFS	Defines the design and language specification in for the project.
PART 6	PSEUDOCODE	The pseudo code would give a rough idea about the how the flow will be and how the functions would be called one after another.
PART 7	APPENDICE	Additional Information and clear-ups about the language in the project.

2. SYSTEM ARCHITECTURE DESCRIPTION

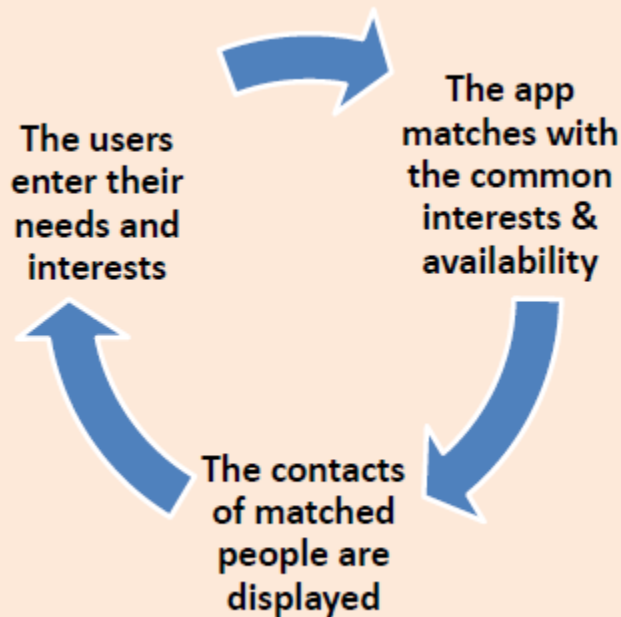
2.1 OVERVIEW OF COMPONENTS

The various components included in this document are:

- Class diagram
- Sequence diagram
- State chart diagram

2.2 STRUCTURE AND RELATIONSHIPS

2.3 USER INTERFACE



The software is cross-platform and can operate on all Windows, Linux and iOS operating systems over all current versions. Since, the software is designed in a social fashion; the app will function fully on android environment.

Design and Implementation Constraints

The app is location specific. Hence, the software requires a mail system login. The privacy policy however provides the users with the authority of whether they want to share their contacts/files on the portal or a specific person. The app is planned to be designed with lower RAM requirements and would require the user to stay online for updates. The software's database is highly secured for privacy purposes. The app/software is designed in English language only and is designed to provide

the users with contact information, not involving any virtual chat station protocols. The programming has been done with html and Java script formats and would provide basic html standards. The app isn't organization based, hence doesn't require any institution's maintenance.

User Documentation

The app is designed to be completely lucid and comprehensible for the users and will be a self start guide within itself.

Assumptions and Dependencies

The app/software is designed keeping in mind the fact that people are willing to share their entertainment measures. There have been surveys performed regarding the same and the results have been positive. The software design is completely original and newly planned for the purpose and the only the language frameworks have been implied for the app/software.

3. DETAILED DESCRIPTION OF COMPONENTS

User Class

Name of Class	User
Attributes	UserId : String Username : String Password : String emailD : String
Methods	Login() : this function takes the details of userid and password and authenticates from database and logs in to the website and the user profile forgotPassword() : this function helps to renew the password of the user in case he/she forgets the password. It is done by sending a mail to the email id of the user.

Name of Class	UserProfile
Attributes	UserId : String Username : String DOB : String About : String emailD : String
Methods	UpdateDetails() : This function helps to update and make changes to any of the details of the user.

Name of Class	Movies
Attributes	Genre : String numMovies : Integer(To keep a count of the number of movies owned by the user)
Methods	AddMovies(movieName,gen) : This function adds a new movie to the database of the user's

	<p>collection of movies.</p> <p>RemoveMovie(movieName) : This function deletes the movies from the user's database which he no longer possesses.</p> <p>CheckMovieDatabase(movieName) : This function checks the database for a movie that is required by the user. If there is someone with the movie, this function gives the contact of that user to the present user so that he can take it.</p>
--	--

Name of Class	Novels
Attributes	<p>Genre : String</p> <p>numBooks : Integer(To keep a count of the number of novels owned by the user)</p>
Methods	<p>AddBook(BookName,gen) : This function adds a new book to the database of the user's collection of books.</p> <p>RemoveBook(BookName) : This function deletes the book(s) from the user's database which he no longer possesses.</p> <p>CheckBookDatabase(BookName) : This function checks the database for a book that is required by the user. If there is someone with the book, this function gives the contact of that user to the present user so that he can take it.</p>

Name of Class	TV Series
Attributes	<p>Genre : String</p> <p>numMovies : Integer(To keep a count of the number of TV Series owned by the user)</p>
Methods	<p>AddSeries(seriesName,gen) : This function adds a new TV Series to the database of the user's collection of movies.</p> <p>RemoveSeries(SeriesName) : This function deletes the TV series from the user's database which he no longer possesses.</p> <p>CheckSeriesDatabase(SeriesName) : This function checks the database for a particular TV Series that is required by the user. If there is someone with the TV Series, this function gives the contact of that user to the present user so that he can take it.</p>

Name of Class	AcademicBooks
Attributes	<p>numMovies : Integer(To keep a count of the number of books owned by the user)</p>
Methods	<p>AddBook(courseCode, book) : This function adds a new book of a particular course with the specific course code to the database of the user's collection of academic books.</p>

	FindBook(coursecode, book) : This function checks whether the book required by the user is present with someone on the database.
--	--

Name of Class	Semester Projects
Methods	AddProject(coursecode,prodetails): Adds to the database all details of a project done in the previous semester. These details on the database can help any junior to do his/her project with ease. FindProject(coursecode): Helps a student to find details about all projects done in a particular course.

4. REUSE AND RELATIONSHIP TO OTHER PRODUCTS

The software is built with the definition of following frameworks:

- 1) The database is designed using the Mongo DB framework.
- 2) The languages (html, CSS, java script, node JS) 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) The Node JS framework: Integration with other Visual Studio features and 3 rd party tools. (Node.js, io.js, JavaScript, TypeScript, HTML, CSS, and JSON support)
- 5) Communication 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

Students living in hostels pass their time mainly in front of laptop screens watching either Hollywood or Bollywood movies, reading novels or hooking up with TV series. Hence, these features were on the top of priority list. This app lets them share their preferences and ask for the same they would like having or watching. There had to be decisions taken in order to derive how exactly will the users share their items.

Several constraints were also taken care of during designing the user interface. The project idea could have faced serious oppositions or failures if the following conditions were somehow invalidated:

- The people are not willing to share their private collections.
- People are not open to interacting with new people or getting to socialize.
- The physical exchange is not possible between two interested parties.
- The people don't sign up or update data or the concerned forum and don't make it visible.

The threats are taken care of in the manner where the app is made completely interactive and lucid for usage. It is considered that students are generally open to socializing and procuring their needed stuff from sources.

The main questions that we tried to answer were:

- What are current process problems? What if the system wasn't implemented?
⇒ The people will have to go on various platforms to find their required leisure time stuff and there are major chances that they don't end up finding them.
- How will the proposed system help?
⇒ The system will provide an easy platform for people to come forward and share their documents on a common portal and even help in socializing.
- What will be the integration problems? Is new technology needed? What skills?
⇒ The integration would require constant update feature and an option to the user to secure their contacts and share only items that they want to. The admin would reserve security rights and the function would be implemented by java-scripting.
- What facilities must be supported by the proposed system?
⇒ The app should be lucid and comprehensible and cross-platform for easy handling and access.

The app then offered facilities ranging uploads and downloads in a manner that a user can easily hold off their contact details and share with only specific users. Also, all options are made available such that the users can navigate through all options.