INDIAN INSTITUTE OF TECHNOLOGY, INDORE

Guest House Booking Platform

Software Requirements Specification (SRS)

BY –

YASH GUTGUTIA, NAMAN JAIN, PRITESH PALOD, PRAKHAR RAI

(180001064) (180001031) (180001038) (180001035)

In partial fulfilment of the requirements

for the degree of

Bachelor of Technology in Computer Science

TABLE OF CONTENTS

4. External Interface Requirements

    - User Interfaces……..
    - Hardware Interfaces……..
    - Software Interfaces…….
    - Communication Interfaces…….

Introduction:

- ## Purpose:

  The purpose of this project is to provide an online booking platform for Guest house accommodation. This document gives an overview of the project and the software requirements for building this platform.

- ## Intended Audience and Reading Suggestions:

  This project has been developed for the students, Faculties, and Administration of the IIT Indore community only. Only IIT Indore members can have access to this facility. This platform has been developed in accordance with the requirements of the hospitality section.

- ## Project Scope:

  The purpose of this guest house booking platform is to ease the process of allocation of accommodation facilities to guardians of IITI students, internship students and (or) guests invited by IITI faculty or administration within the campus.

- ## References:

  - https://flutter.dev/
  - http://fluttericon.com/
  - http://dart.dev

Overall Description:

- ## Product Perspective:

The guest house booking database will include:
Customer description: It will contain all the details related to customer which maybe further used for various purposes.
Guest house room description: It will contain the details related to guest houses and the rooms in those guest houses.
Booking description: It will contain the details related to all the bookings that are done like the referring person, date of visit, room no., etc.

- ## User Class and Characteristics:

Users should be able to retrieve information about the availability of rooms for booking for particular dates and time.
There are two types of users – Customers and Admin management.
Customer functions:

- Book rooms
  - For particular days
  - In different guest houses
  - Confirmation
  - Payment

Hospitality management functions:

- Get the details of all the bookings that have been done.
- Get the data related to each room (availability status)
- To verify the referencing person
- Details of customer and authorizing dignitary
- Update booking prices

- Operating environment:
    - Centralized database
    - Client server system
    - Operating system: Android, iOS
    - Database: Firebase
    - Platform: Android Studio using Dart

System Features:

- ## Description and Priority:

  This guest house booking platform maintains records of guests coming to institute along with other information related to bookings such as check-in date and check-out date. This platform also allows retrieval of information related to the availability of rooms for a specific period of time.
  This project has a high priority because only IITI administration or IITI faculty members or IITI students (for their relatives) are allowed to book rooms.
  The main motivation behind this project is to reduce the paper work and other tedious jobs hospitality section had to do in absence of such a platform for guest house booking.

- ## Stimulus/Response Sequences:

  - Login/Sign up as Student/Admin/Teaching and non-teaching staff of IITI
  - Check room availability status for various guest houses
  - Book room in any guest house according to preference and availability
  - Get confirmation of the booking
  - Pay the required amount

- ## Functional Requirements:

  - Centralized Database:

    A centralized database (sometimes abbreviated CDB) is a database that is located, stored, and maintained in a single location. This location is most often a central computer or

database system, for example a desktop or server CPU, or a mainframe computer. In most cases, a centralized database would be used by an organization (e.g. a business company) or an institution (e.g. a university.) Users access a centralized database through a computer network which is able to give them access to the central CPU, which in turn maintains to the database itself.

- Client/server system:

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).
A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

External Interface Requirements:

- User Interfaces:
    - Front End Software-
      Dart:

      Dart is a client-optimized language for fast apps for any platform.
      Optimized for UI:
        - Mature and complete async-await for user interfaces containing event-driven code, paired with isolate-based concurrency.
        - A programming language optimized for building user interfaces with features such as the spread operator for expanding collections, and collection if for customizing UI for each platform.
        - A programming language that is easy to learn, with a familiar syntax.

      Productive development:

        - Make changes to your source code iteratively, using hot reload to instantly see the effect in the running app
        - Write code using a flexible type system with rich static analysis and powerful, configurable tooling
        - Do profiling, logging, and debugging with your code editor of choice
          Fast on all platforms:
        - AOT-compile apps to native machine code for instant start-up
        - Target the web with complete, mature, fast compilers for JavaScript
        - Run backend code supporting your app, written using a single programming language

Flutter:

Flutter is Google’s UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase.
Fast development:

Flutter's hot reload helps you quickly and easily experiment, build UIs, add features, and fix bugs faster. Experience sub-second reload times, without losing state, on emulators, simulators, and hardware for iOS and Android.

Expressive, beautiful UIs:

Delight your users with Flutter's built-in beautiful Material Design and Cupertino (iOS-flavour) widgets, rich motion APIs, smooth natural scrolling, and platform awareness.

Native Performance:

Flutter’s widgets incorporate all critical platform differences such as scrolling, navigation, icons and fonts to provide full native performance on both iOS and Android.

- Back End Software-

Firebase:

Build apps fast, without managing infrastructure:

Firebase gives you functionality like analytics, databases, messaging and crash reporting so you can move quickly and focus on your users.

Backed by Google, trusted by top apps:

Firebase is built on Google infrastructure and scales automatically, for even the largest apps.

One platform, with products that work better together:

Firebase products work great individually but share data and insights, so they work even better together.

- Hardware Interfaces:

    - Android/iOS (for deployment)
    - Windows (for development)

- Software Interfaces:

    - Operating System – The app will be available for both Android and iOS because of its wide usage and smooth interface.
    - Database – Firebase for Centralized Database.
    - Dart – We have chosen to work upon Dart as our front-end language because it supports Flutter, which contains wide range of widgets, and being a cross-platform IDE, will allow us to create an app for both Android and iOS without using two native languages.

- Communication Interfaces:

    We are using simple electronic forms for the booking rooms and collecting other information.