ATMIYA UNIVERSITY

RAJKOT

A

Report On

Resume builder

Under subject of

MINI PROJECT

B.Tech., Semester – VII

(Computer Engineering)

Submitted by:

Drashti Ranpariya (220007006)

Ms. Tosal M. Bhalodia

(Faculty Guide)

Ms. Tosal M. Bhalodia

(Head of the Department)

Academic Year

(2025-26)

CANDIDATES'S DECLARATION

I hereby declare that the work presented in this project entitled "**Resume Builder**" submitted towards completion of project in **7th Semester** of B. Tech. (Computer Engineering) is an authentic record of our original work carried out under the guidance of "Ms. Tosal M. Bhalodia".

I further declare that this project has not been submitted by me for the award of any other degree, diploma, or certificate in any other institution. The results, features, and implementation described in this project are based on my own effort and research.

Semester: VII

Place: Rajkot

Signature:

Drashti Ranpariya (220007006)

ATMIYA UNIVERSITY

RAJKOT

CERTIFICATE

Date:

This is to certify that the "**Resume builder**" has been carried out by **Drashti Ranpariya** under my guidance in fulfilment of the subject Mini Project in COMPUTER ENGINEERING (7th Semester) of Atmiya University, Rajkot during the academic year 2025-26.

Ms. Tosal M. Bhalodia **Ms. Tosal M. Bhalodia**

(Project Guide) **(Head of the Department)**

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my guide **Ms. Tosal M. Bhalodia**, for their invaluable guidance, encouragement, and support throughout the development of this mini project. Their constant motivation and valuable insights helped me to complete this work successfully.

I also extend my heartfelt thanks to the **Head of Department Ms. Tosal M. Bhalodia** and all faculty members of the **Department of Computer Engineering Atmiya University** for providing me with the necessary resources and a conducive environment for completing this project.

Finally, I am deeply thankful to my friends and family for their continuous support, cooperation, and encouragement during the course of this project.

Drashti Ranpariya (220007006)

ABSTRACT

A resume is an essential document for individuals in the academic, professional, and corporate worlds. Crafting a resume manually is often time-consuming and prone to formatting inconsistencies. The Resume Builder Application seeks to automate this process by providing a structured, user-friendly, and efficient tool that enables users to create professional resumes seamlessly.

This project utilizes Flutter, a cross-platform UI toolkit developed by Google, which allows a single codebase to deploy the application on both Android and iOS platforms. The core functionality includes structured form-based data collection, real-time validation, modular storage using a Singleton design pattern, and the ability to generate resumes dynamically using multiple customizable templates.

Key features include:

- Section-wise data input (Personal Details, Education, Skills, Experience, Projects, Certifications, Awards, and Hobbies).
- Validation to ensure completeness and correctness of user data.
- Multiple professional resume templates for customization.
- Real-time preview and editing capabilities.
- Export options such as PDF generation for sharing or printing.

The project is highly scalable and has potential future enhancements such as cloud integration, AI-based resume suggestions, and job portal integrations.

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1 Purpose

The purpose of the Resume Builder Application is to simplify and digitize the resume creation process. Traditionally, individuals rely on word processors like Microsoft Word, Google Docs, or online resume websites, but these methods require formatting knowledge and are often tedious. This project aims to:

- Eliminate the need for manual formatting.
- Offer ready-to-use professional templates.
- Provide an interactive and user-friendly interface for resume creation.
- Reduce time and effort in resume preparation.
- Ensure data accuracy through structured validation mechanisms.

1.2 Scope

The application targets students, freshers, job seekers, and professionals who need resumes tailored to specific opportunities. By supporting multiple templates, the application allows customization for different fields (technical jobs, creative roles, academic positions, etc.).
Key scope areas include:

- **Cross-platform availability:** The app runs on Android and iOS.
- **Data modularity:** Each section of the resume is handled independently.
- **User privacy:** All data is stored locally; no external server communication.
- **Scalability:** Future upgrades may include cloud storage, AI-based suggestions, and template marketplace.

1.3 Technology and Tools

- Frontend & Backend: Flutter (Dart)
- IDE: Android Studio / Visual Studio Code
- Database: Local storage (Singleton pattern + in-memory structures)

Libraries:

- provider for state management
- pdf for resume generation
- image_picker for profile image support
- google_fonts for typography

CHAPTER-2

PROJECT MANAGEMENT

2.1 Project Planning

The development followed a modular approach. The project was divided into phases to ensure smooth progress:

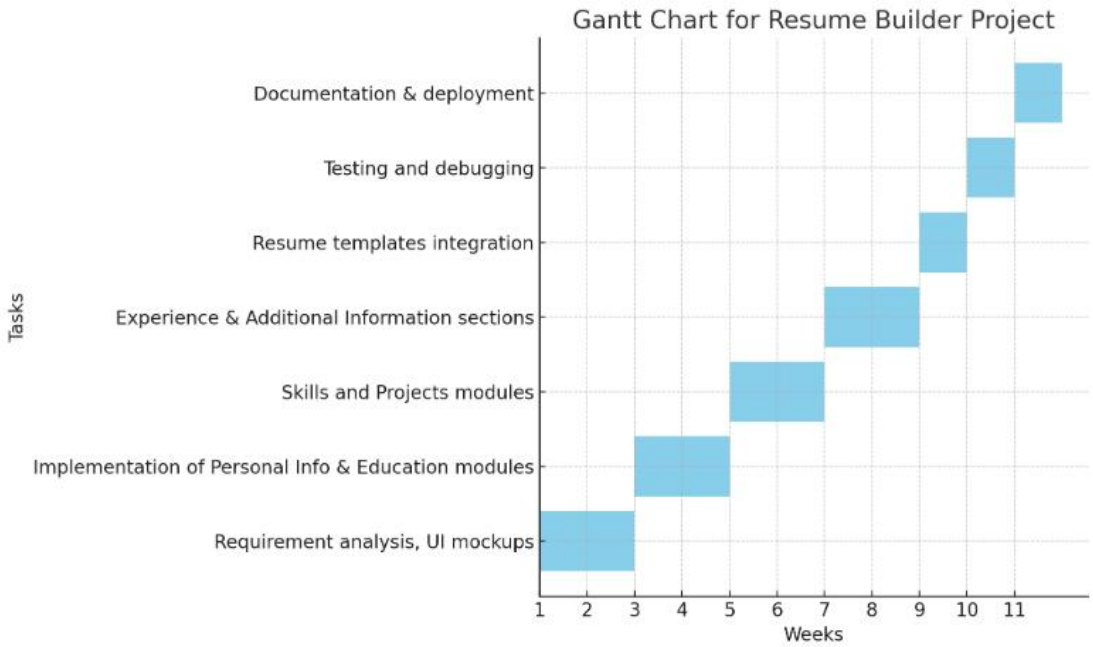
1. Requirement Gathering: Identifying user requirements for resume sections.
2. System Design: Creating UML diagrams, data models, and planning user flows.
3. Module Development: Implementing each section like Personal Info, Education, etc.
4. Integration: Combining modules and ensuring smooth data flow.
5. Testing: Validating functionality, form validation, and template rendering.
6. Deployment: Packaging the app for Android and iOS.

2.2 Project Scheduling

The Agile methodology was followed with weekly sprints. Each sprint covered one or more features/modules. Example schedule:

Gantt chart: It is a project management tool that visually represents tasks against a timeline, showing their start and end dates to track progress and scheduling.

Figure 2.2.1 Gnatt Chart for Resume Builder App



- Week 1–2: Requirement analysis, UI mockups.
- Week 3–4: Implementation of Personal Info & Education modules.
- Week 5–6: Skills and Projects modules.
- Week 7–8: Experience & Additional Information sections.
- Week 9: Resume templates integration.
- Week 10: Testing and debugging.
- Week 11: Documentation & deployment.

2.3 Risk Management

2.3.1 Risk Identification:

Technical Risks:

- Flutter SDK updates breaking package compatibility.
- PDF rendering errors across devices.

Operational Risks:

- Incomplete user data leading to broken templates.
- Users demanding additional templates beyond initial scope.

Security Risks:

- Unauthorized data sharing if cloud integration is added in the future.

2.3.2 Risk Analysis

Table 2.3.2 Risk Analysis

Impact Level	Risk	Mitigation
High	SDK/package compatibility	Use stable versions of SDKs and packages.
Medium	Data loss due to local-only storage	Implement cloud backup in the future.
Low	UI bugs due to device screen variations	Use a responsive layout in Flutter.

CHAPTER-3

SYSTEM REQUIREMENTS STUDY

3.1 Hardware and Software Requirement

This shows minimum requirements to carry on to run this system efficiently.

3.1.1 Hardware Requirements:

Table 3.1.1 Server-side Hardware Requirement

Category	Requirement
Processor	Minimum Quad-core processor
RAM	16 GB RAM for efficient multitasking
Storage	At least 500 GB SSD storage
Internet & Hosting	High-speed internet with stable hosting infrastructure
OS	Linux (Ubuntu/RedHat) or Windows Server

3.1.2 Software Requirements:

Table 3.1.1 Software Requirements

Category	Requirement
SDK	Flutter SDK (3.x or above) for cross-platform development
Programming Language	Dart Programming Language as the primary coding language
IDE	Android Studio / VS Code as Integrated Development Environment (IDE)
Version Control	Git & GitHub
Packages	PDF package (pdf Flutter library), Image Picker package
Typography	Google Fonts for elegant typography integration

3.1.3 Minimum Client Device Requirements:

3.1.3 Minimum Client Device Requirements

Category	Requirement
Android OS	Version 8.0 (Oreo) and above
iOS	Version 12.0 and above
Processor	Quad-core CPU or higher
RAM	Minimum 2 GB, recommended 4 GB
Storage	50 MB free space for installation, additional space for PDF export

Test Device Used:

- Android 14, 128 GB storage, 4 GB RAM

3.2 Constraints

3.2.1 Hardware Limitations:

Older smartphones with limited processing capacity or small screen sizes may face minor performance or UI rendering issues. This limitation can be resolved by ensuring responsive layouts and optimized assets in the app.

3.2.2 Reliability Requirements:

The application must reliably save user-entered data without accidental loss. ResumeData (singleton class) ensures consistent access to all user information throughout the application lifecycle.

3.2.3 Safety and Security Considerations:

- Data is stored locally to ensure privacy.
- No third-party server calls are made, avoiding risks of unauthorized data sharing.
- For future cloud integration, encryption protocols (e.g., HTTPS, SSL/TLS) and secure authentication mechanisms will be considered.

CHAPTER-4

SYSTEM ANALYSIS

4.1 Study Current System

Currently, resumes are built using:

1. **Manual Methods:** Tools like Microsoft Word, Google Docs, or LaTeX require formatting expertise.
2. **Online Builders:** Websites like Canva or Zety offer templates but are often paid or require internet connectivity.
3. **Pre-designed Templates:** Downloadable templates lack customization flexibility.

4.2 Problem and weakness of current system

- **Time-Consuming:** Manual formatting in Word/Docs takes significant effort.
- **Limited Access:** Online builders need stable internet.
- **High Cost:** Premium templates on third-party platforms are not free.
- **Inconsistency:** Non-standard formatting leads to poorly structured resumes.

4.3 Requirements of New System

4.3.1 User Requirements:

- Easy-to-use mobile app with intuitive UI.
- Section-based input (Personal, Education, Skills, etc.).
- Option to preview before export.
- Multiple templates to suit industry-specific resumes.

4.3.2 System Requirements:

- Efficient memory usage (lightweight storage).
- Smooth data transfer across different screens.
- Responsive layout for multiple device sizes.

4.4 Feasibility Study

- **Technical Feasibility:** Flutter ensures cross-platform support. Packages for PDF generation and image handling are well-documented.

- **Economic Feasibility:** No additional cost as it uses open-source tools.
- **Operational Feasibility:** End-users can learn quickly; no technical knowledge required.

4.5 Feature Of New System

- Structured resume creation process.
- Real-time validation (required fields).
- Multiple templates (Template 1, 2, 3, and 4 with image).
- Export resume in PDF format.
- Modern UI with animations for professional appeal.

CHAPTER-5

SYSTEM DESIGN

5.1 Input / output Interface

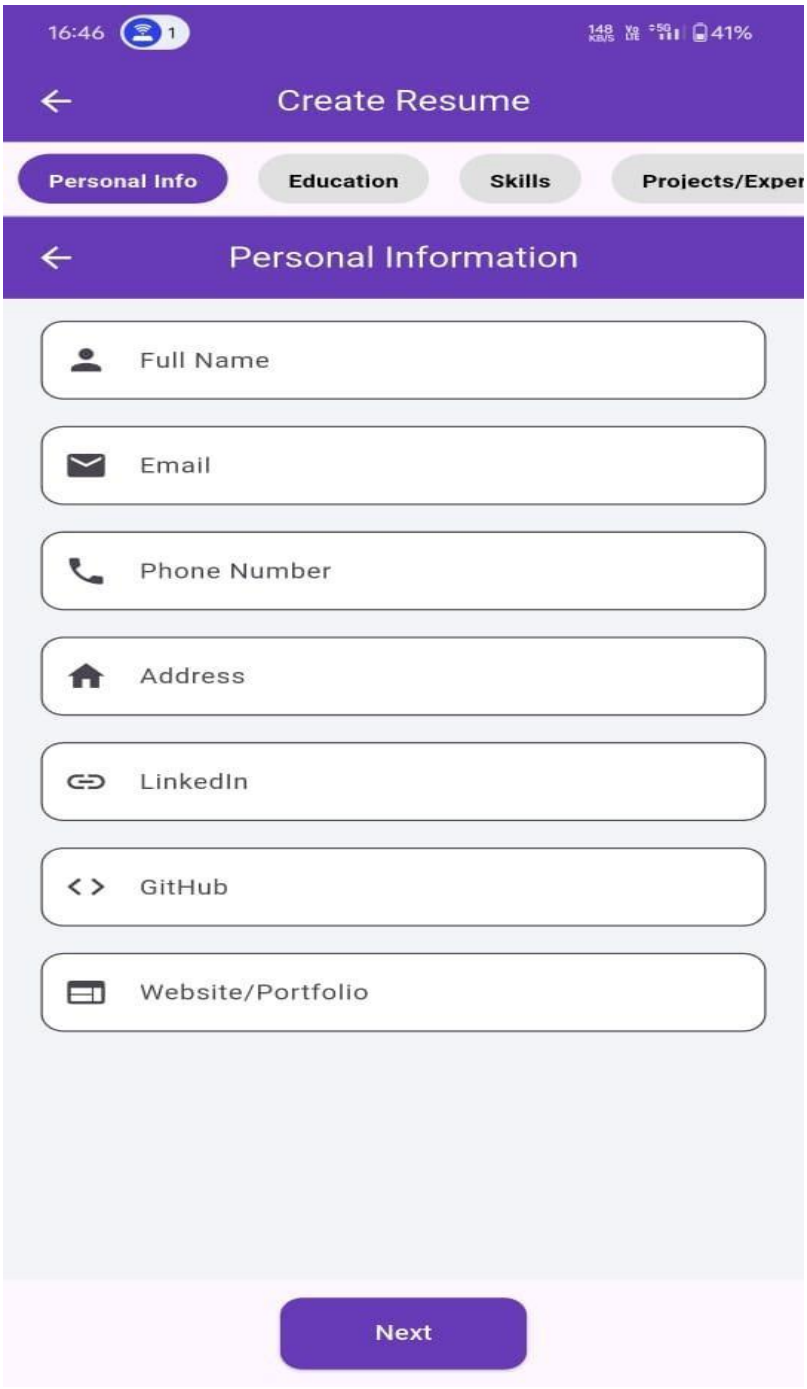


Figure 5.1.1 Personal Information Screen

Personal Information input screen where users enter basic details such as name, email, phone, and social links for resume creation.

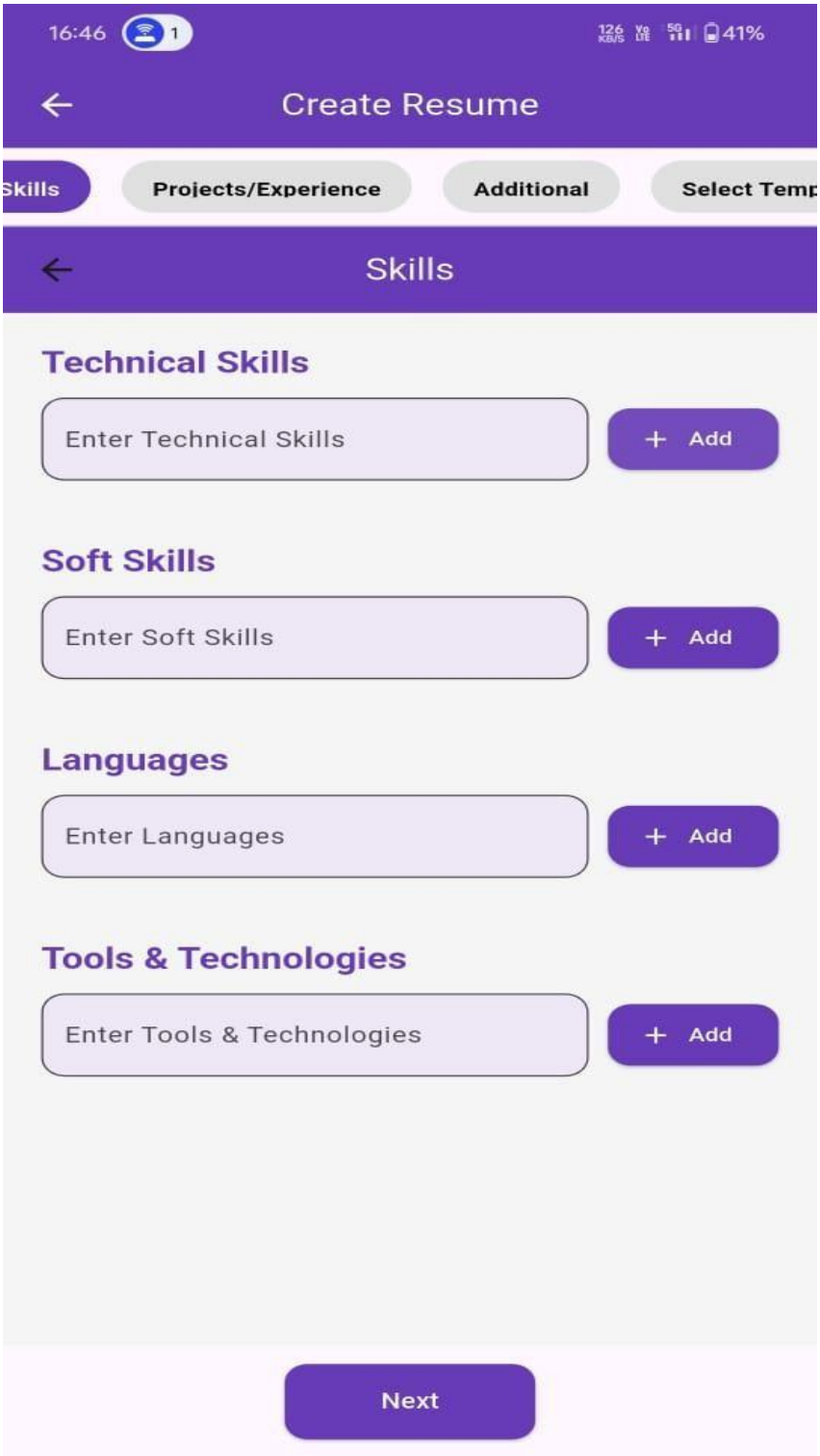


Figure 5.1.2 Skills Screen

Skills entry screen allowing users to add technical skills, soft skills, languages, and tools/technologies for their resume.

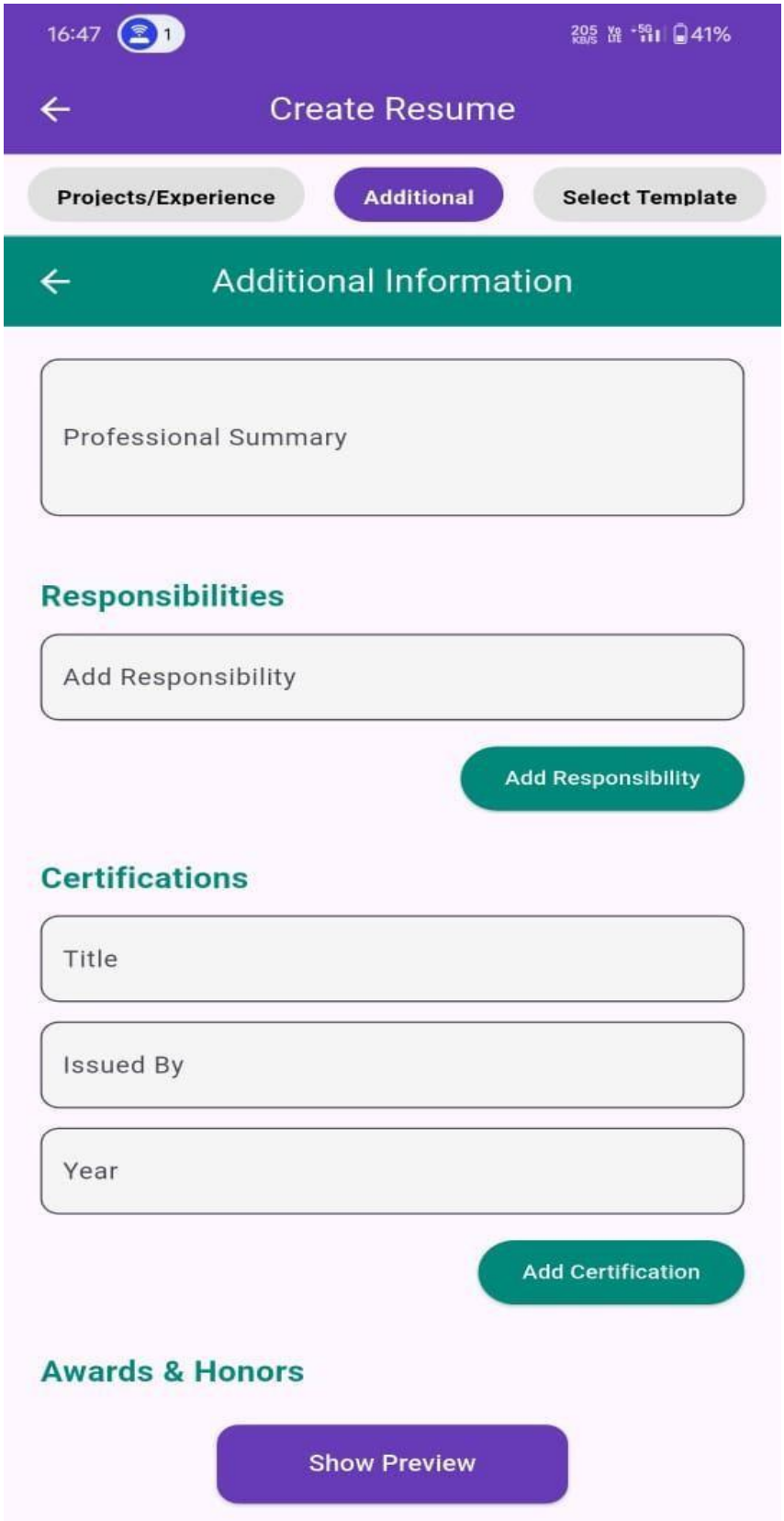


Figure 5.1.3 Additional Information Screen

Additional Information screen where users can add a professional summary, responsibilities, certifications, and awards for their resume.

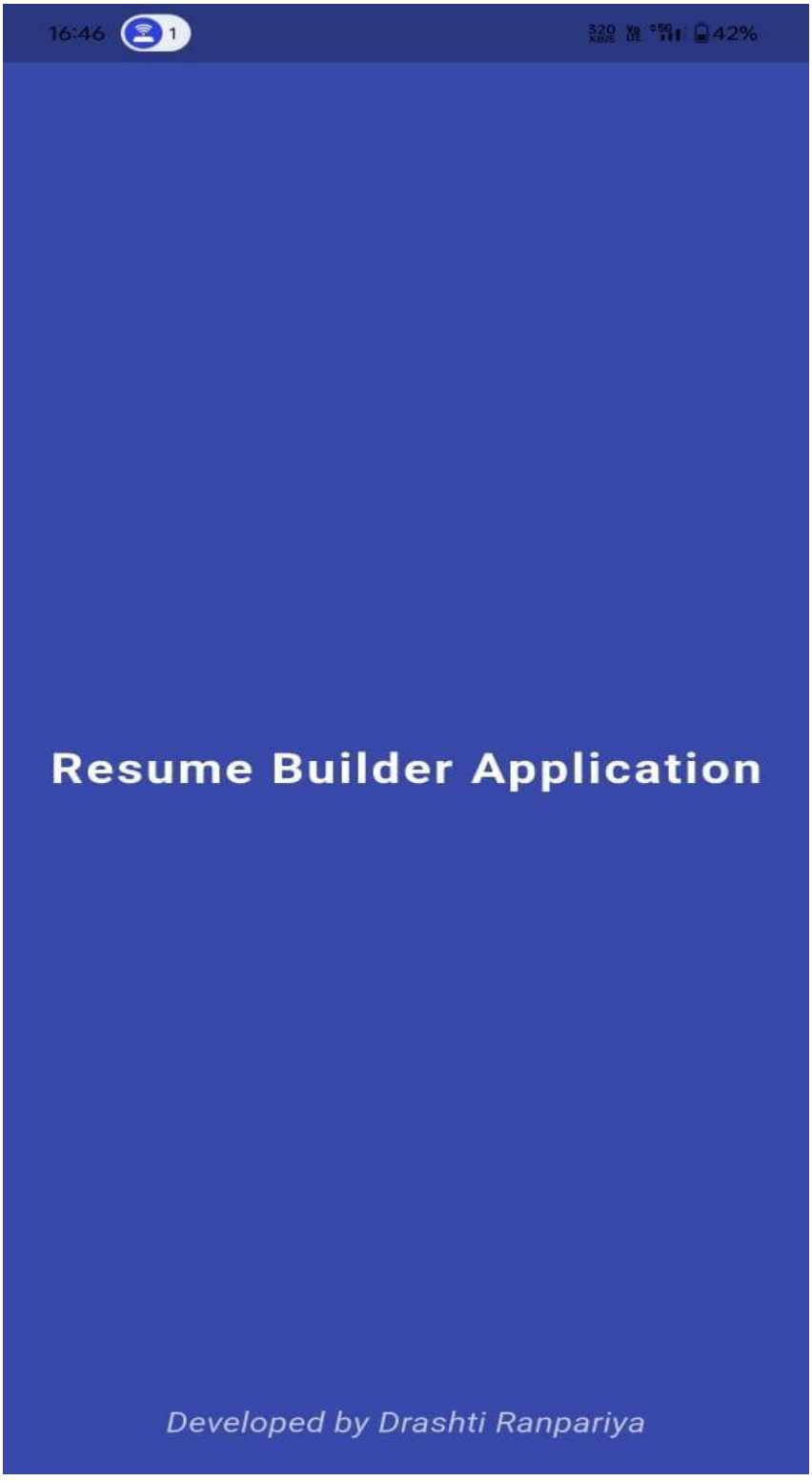


Figure 5.1.4 Splash Screen

Splash screen of the Resume Builder Application displaying the app title and developer name

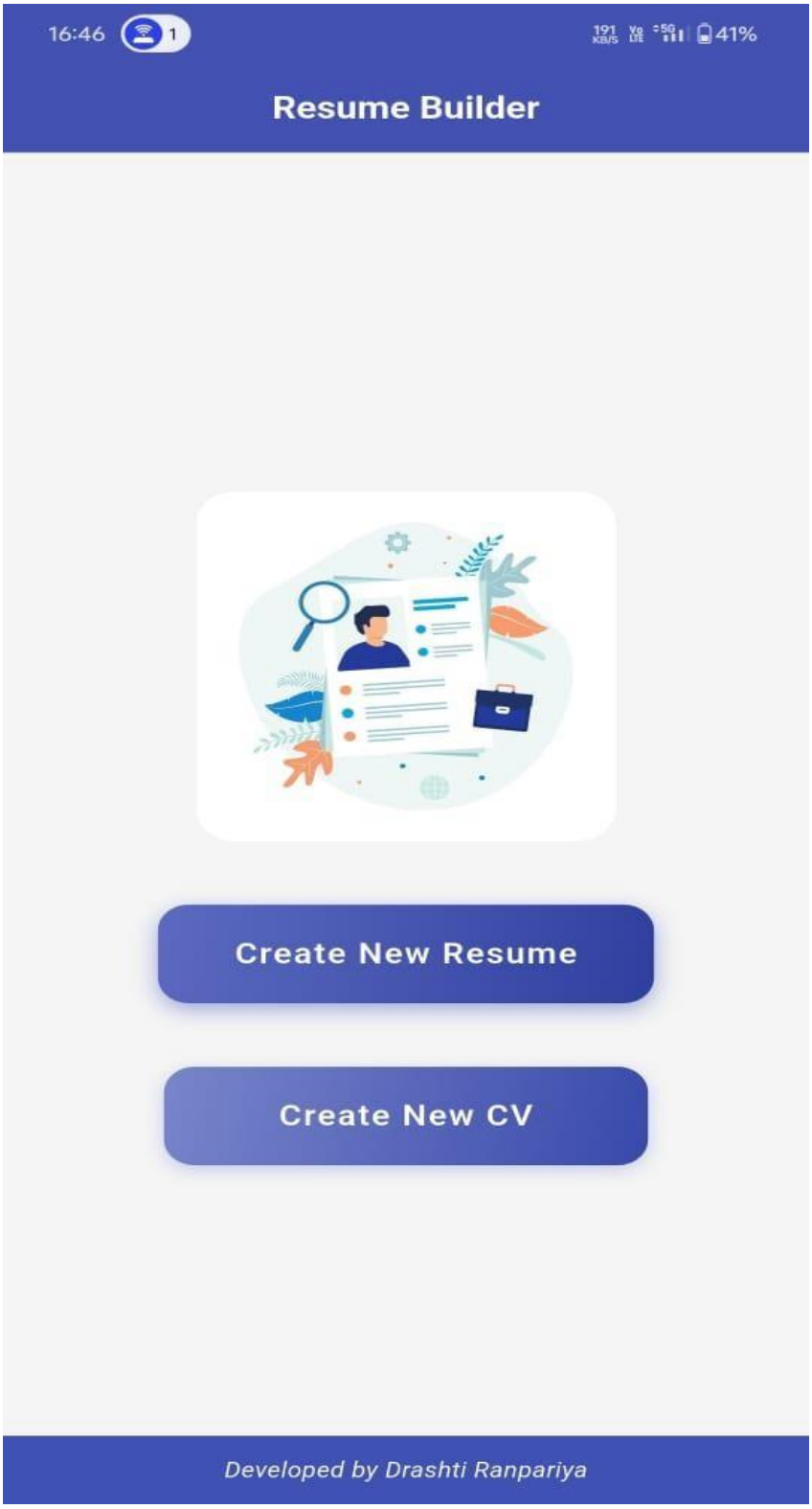


Figure 5.1.5 Home Screen

Home screen of the Resume Builder Application with options to create a new resume or CV.

Figure 5.1.6 CV Template

CV template selection screen where users can choose from multiple colorful templates

Figure 5.1.7 Resume Template

Resume template selection screen where users can choose from multiple colorful templates

5.2 Interface Design

5.2.1 Class Diagram:

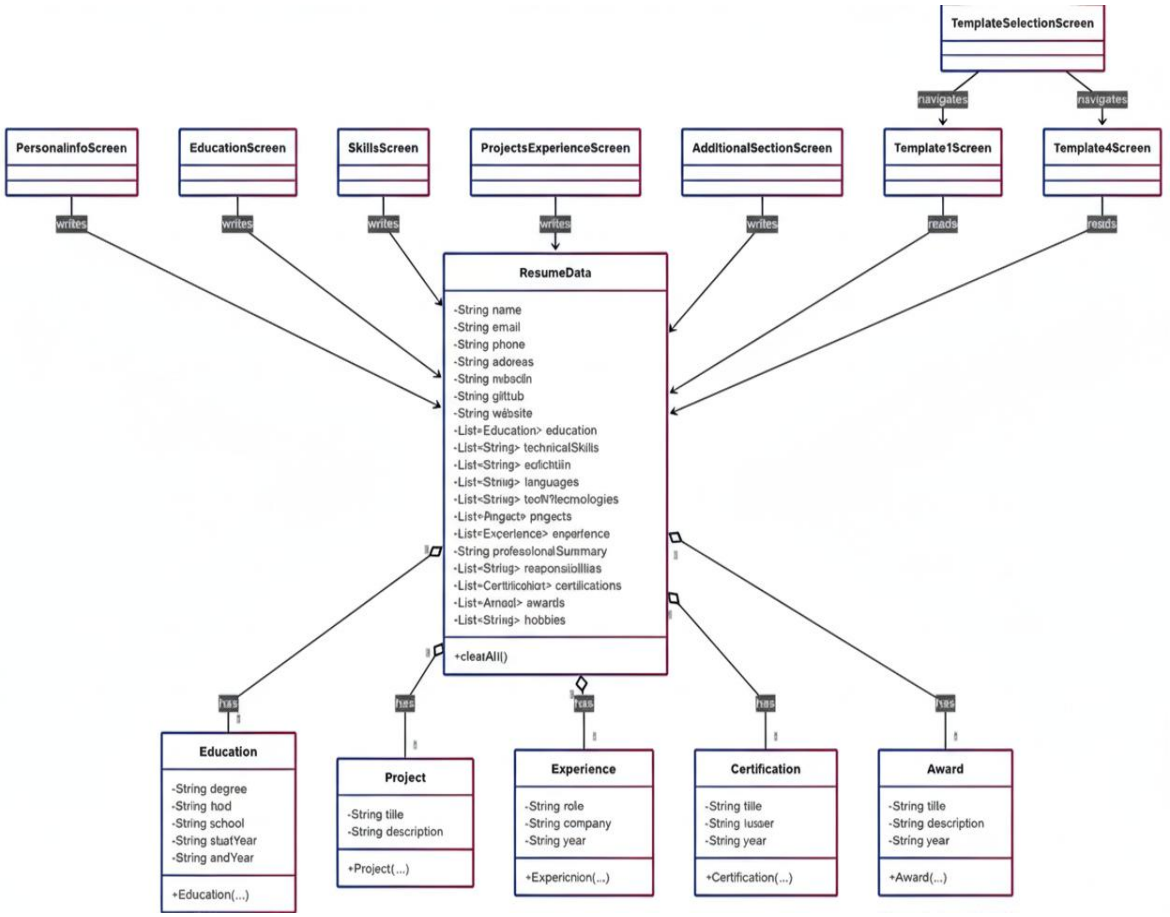


Figure 5.2.1 Class Diagram

Class diagram: It shows the main ResumeData class as a central hub that contains attributes for various personal and professional details, and has one-to-many relationships with other classes like Education, Project, Experience, Certification, and Award to store detailed, multiple entries.

5.2.2 Use Case Diagram:

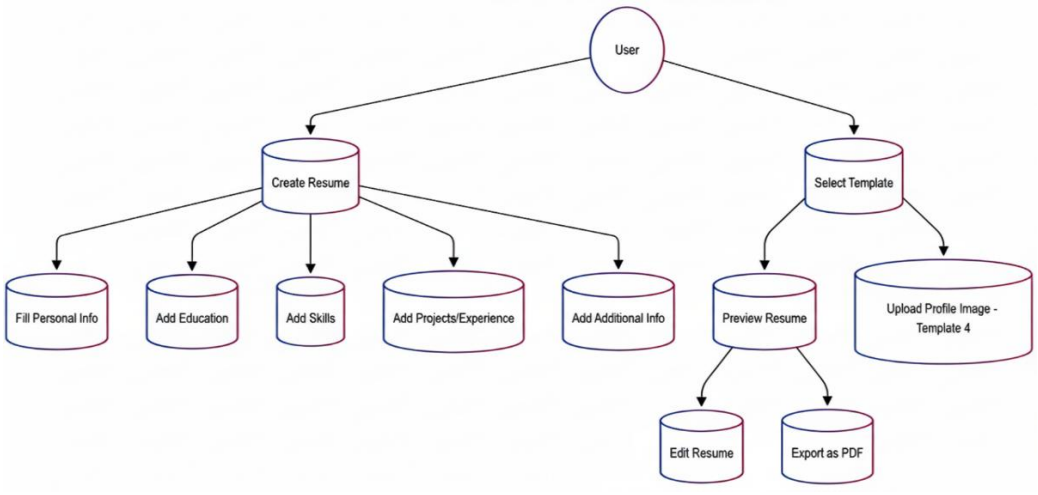


Figure 5.2.2 Use-case diagram

Use-Case: This is a use case diagram showing the core functionalities of a resume builder app. It illustrates how a User can Create a Resume by filling out different sections, as well as Select a Template, preview the result, and finally Export as PDF.

5.2.3 Activity Diagram:

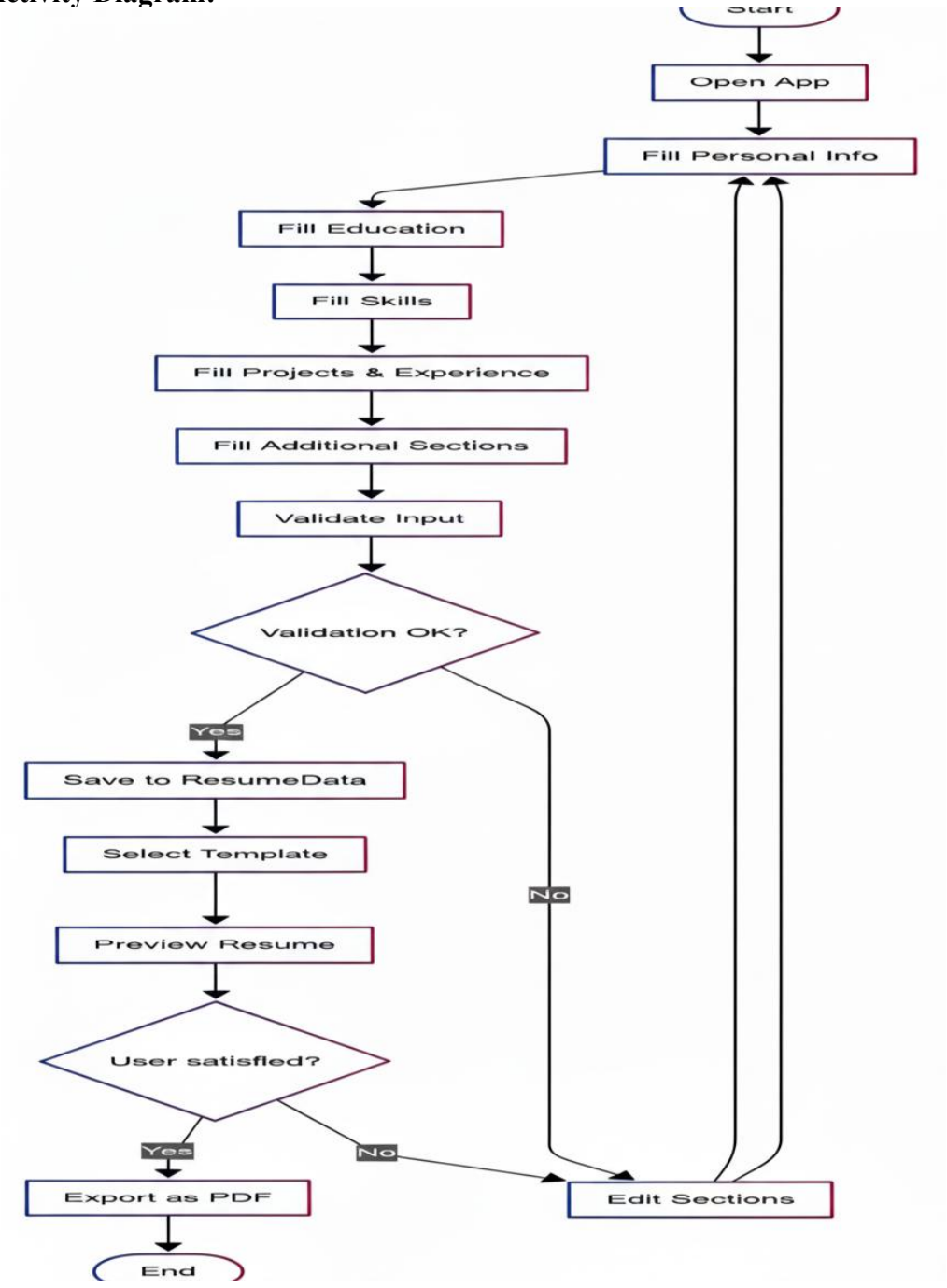


Figure 5.2.3 Activity Diagram

Activity: This activity diagram shows the user flow for a resume builder application, starting with opening the app and filling in personal information. The process includes a validation step that loops back to allow edits if needed, and a final check before exporting to PDF that also allows for revisions.

5.2.4 Data Flow Diagram:

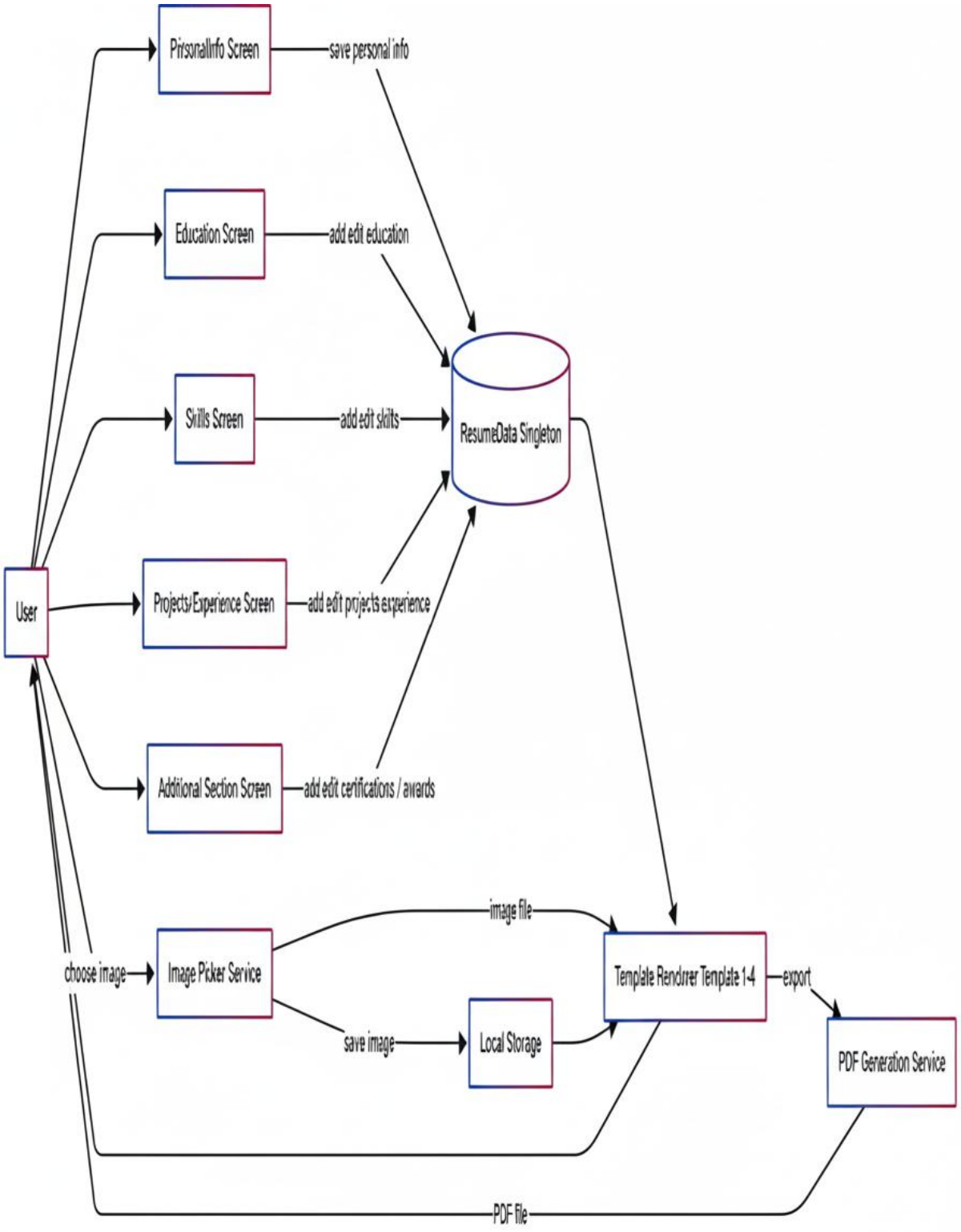


Figure 5.2.4 Data Flow Diagram

Data Flow Diagram: This is a data flow diagram that shows how data moves through a resume builder application. It illustrates how user input from different screens is saved to a central ResumeData Singleton and then used to render a resume template, which can be exported as a PDF.

5.2.5 State Diagram:

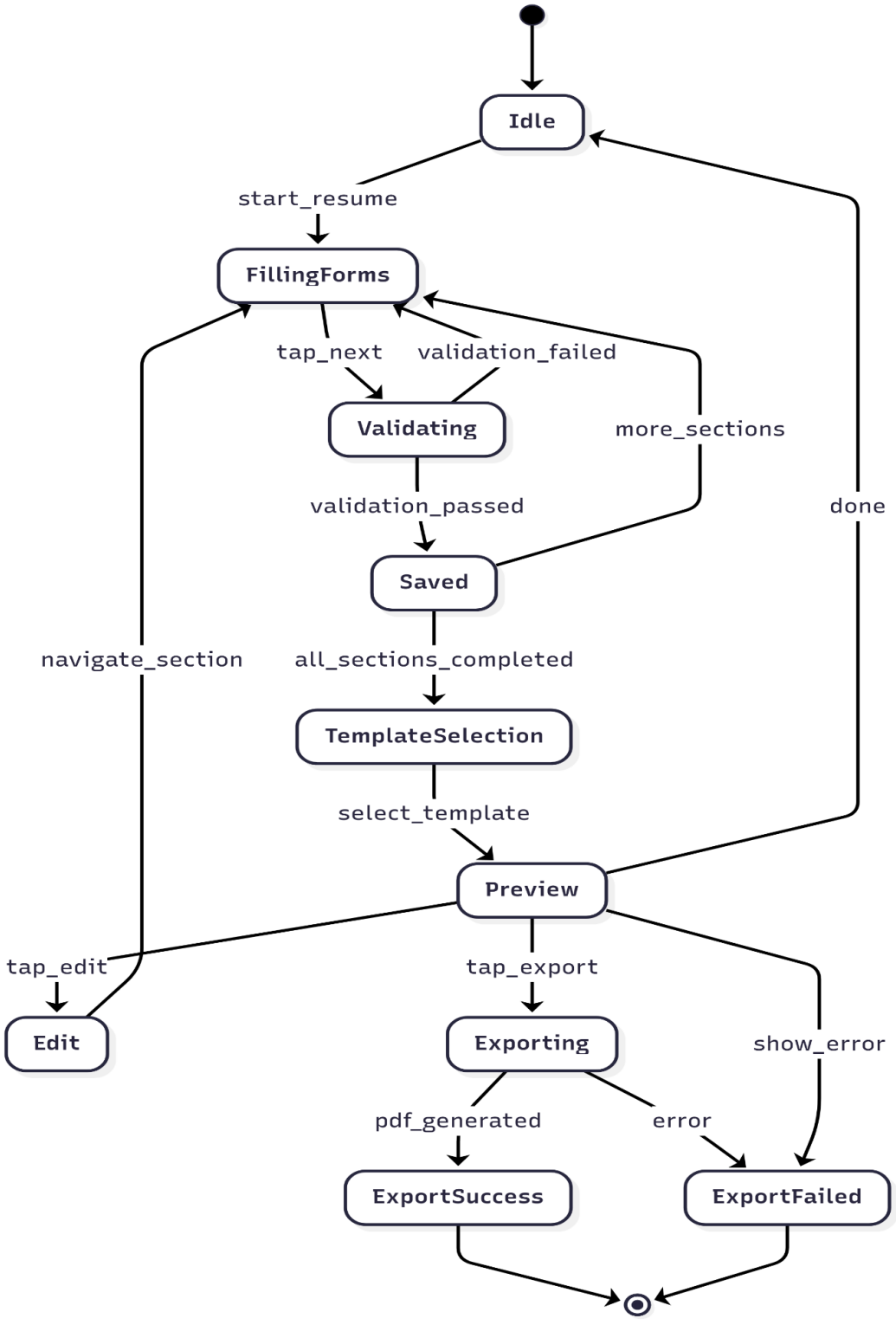


Figure 5.2.5 State Diagram

State Diagram: This state diagram shows the different states of a resume builder application, starting from an **Idle** state. It transitions through states like **FillingForms**, **Validating**, and **Exporting** as the user interacts with the app, with specific events like validation_failed or tap_edit triggering state changes.

5.2.6 E-R Diagram:

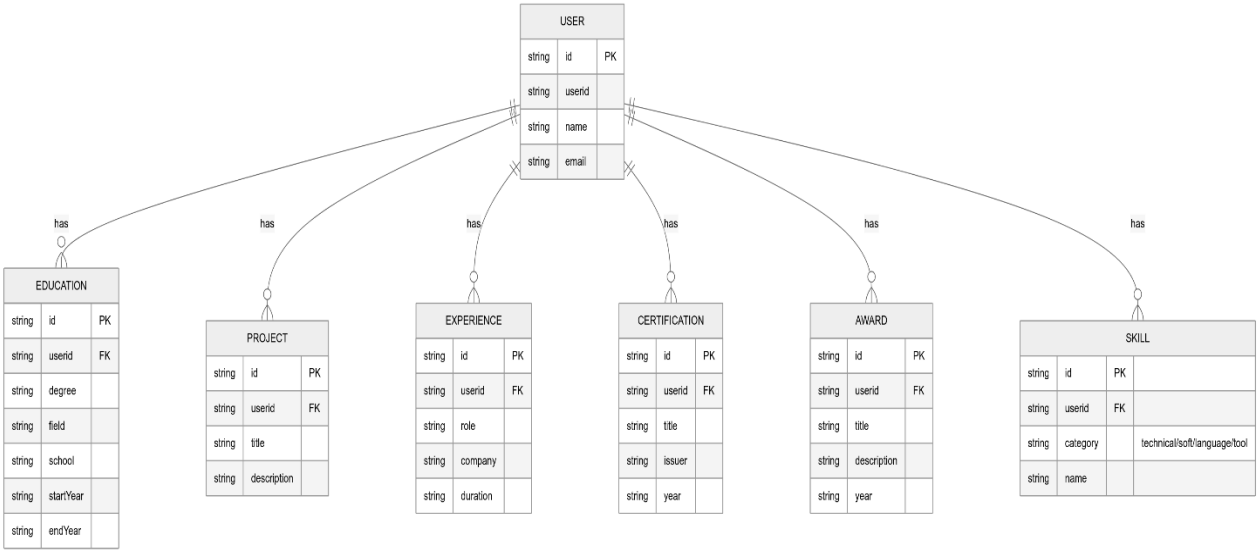


Figure 5.2.6 E-R Diagram

E-R Diagram: It shows a central USER entity connected to various data entities like EDUCATION, PROJECT, and EXPERIENCE, illustrating how a single user can be associated with multiple entries in each category.

5.2.7 Sequence Diagram:

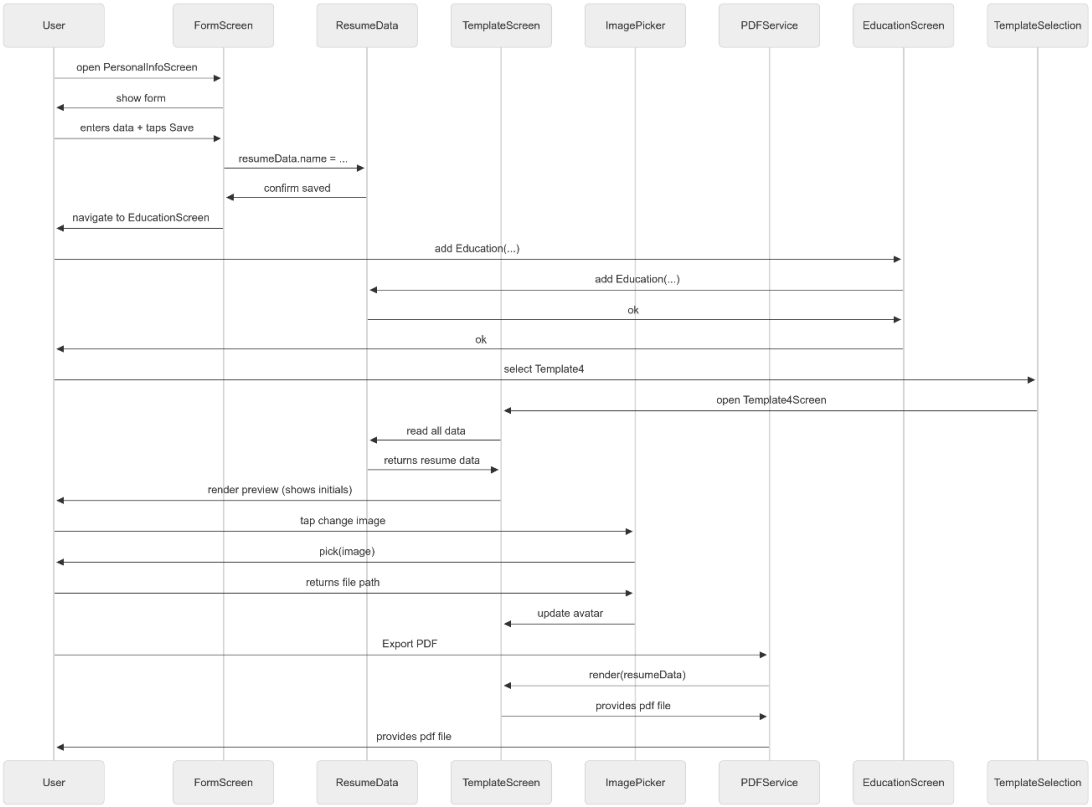


Figure 5.2.7 Sequence Diagram

Sequence Diagram: This is a sequence diagram showing the interactions between a User, different screens, and various services as they build a resume, from filling out forms to exporting a PDF.

CHAPTER-6

CODE IMPLEMENTATION

6.1 Implementation Environment :

- Dart language
- Flutter framework
- Android Studio / VS Code IDE

6.2 Program / Module Specification :

- Personal Info Screen: Captures name, email, phone, etc.
- Education Screen: Captures degree, school, start/end year.
- Skills Screen: Captures technical & soft skills.
- Projects Experience Screen: Captures project & work experience details.
- Additional Section Screen: Captures certifications, awards, hobbies, summary.
- Template Screens: Render resume layouts.

6.3 Coding Standards :

- Consistent naming conventions (camelCase).
- Modular widget design for reusability.
- Use of Provider/Singleton pattern for data management.
- Error handling for null/empty data.

CHAPTER-7

TESTING

Testing is a critical stage in software engineering to ensure that the developed application meets user requirements, functions correctly, and performs reliably under different conditions. For the Resume Builder Application, multiple testing strategies and methods were applied.

7.1 Testing Strategy :

The primary strategy followed was a combination of Black-Box Testing and White-Box Testing:

- **Black-Box Testing:** Focused on checking functionalities like form validation, navigation between sections, template rendering, and PDF export without considering the internal logic.
- **White-Box Testing:** Verified logical flow within classes like ResumeData, ensuring that data persistence and retrieval functions were working as expected.

Additionally, Regression Testing was applied after integrating new features (like Template 4 with image upload) to ensure previous functionalities still worked correctly.

7.2 Testing Methods

7.2.1 Unit Testing

Each module was tested independently:

- Personal Info Screen: Ensured required fields (Name, Email, Phone) must be filled.
- Education Screen: Verified multiple entries could be added and deleted.
- Skills Screen: Confirmed duplicate skills could not be added.
- Projects Experience Screen: Validated proper addition and removal of projects / experience.
- Template Screens: Verified dynamic rendering of data.

7.2.2 Integration Testing :

Integration testing checked the seamless flow between modules:

- Data entered in PersonalInfoScreen successfully reflected in Template 1–4.
- Switching between SkillsScreen → ProjectsExperienceScreen → AdditionalSectionScreen retained entered data.

- Profile image uploaded in Template 4 persisted even after reopening the template.

7.2.3 Validation Testing

- Ensured email format must include @.
- Phone number must be numeric with valid length.
- Mandatory fields (e.g., Name, Degree) cannot be left blank.
- Prevented users from proceeding to next screen if validations failed.

7.3 Test Cases :

Table 7.3.1 Test Suite

Test Case ID	Module	Input	Expected Output	Result
TC-01	Personal Info	Empty Name field	Error: "Name is required"	Pass
TC-02	Education	Add Degree + School details	Entry saved to ResumeData	Pass
TC-03	Skills	Add duplicate skill "C++"	Only one entry saved	Pass
TC-04	Projects	Add project with description	Project displayed in Template	Pass
TC-05	Experience	Remove entry	Entry successfully removed	Pass
TC-06	Template Selection	Choose Template 1	Template 1 loads with data	Pass
TC-07	Template 4	No image uploaded	Initial avatar with first letter displayed	Pass
TC-08	Template 4	Upload profile image	Image replaces avatar and is persistent	Pass
TC-09	Export	Export Resume as PDF	PDF generated without data loss	Pass

CHAPTER-8

LIMITATIONS AND FUTURE ENHANCEMENT

8.1 Limitations :

- **Limited Templates**: Currently, only 4 templates are supported. More variety is needed to cater to different industries (creative resumes, minimalist layouts, etc.).
- **Local Storage Only**: Data is stored locally; uninstalling the app removes all resumes.
- **Device Dependency**: Older smartphones with <2 GB RAM may face lag during PDF export.
- **No Cloud Sync**: Users cannot access resumes across multiple devices.
- **No Multi-language Support**: App currently supports English only.

Table 8.1 Limitations

Category	Limitation/Challenge
Templates	Limited to 4 templates; lacks variety for different industries (e.g., creative, minimalist).
Data Storage	Local storage only; uninstalling the app results in permanent data loss.
Performance	Lagging during PDF export on older smartphones with less than 2 GB of RAM.
Cloud Sync	No ability to access resumes across multiple devices.
Localization	Currently supports English only; lacks multi-language support.

8.2 Future Enhancements :

- Cloud Integration: Allow saving resumes in Google Drive or Dropbox.
- AI Resume Suggestions: Suggest improvements in phrasing, grammar, and keyword optimization for ATS (Applicant Tracking Systems).
- Template Marketplace: Allow downloading/buying more templates within the app.
- Multi-language Resumes: Support for multiple languages like Hindi, Spanish, French.
- Sharing Options: Share resumes directly via email, LinkedIn, or WhatsApp.
- Dark Mode Support: Modern UI trend for better user experience.
- Web Version: Extend app to web using Flutter Web for desktop usage.

Table 8.2 Future Enhancements

Enhancement	Description
Cloud Integration	Allows users to save resumes to cloud services like Google Drive or Dropbox.
AI Resume Suggestions	Provides AI-powered recommendations for improving resume content, grammar, and keyword optimization for Applicant Tracking Systems (ATS).
Template Marketplace	Enables users to download or purchase additional resume templates from within the app.
Multi-language Resumes	Adds support for creating resumes in multiple languages, such as Hindi, Spanish, and French.
Enhanced Sharing Options	Allows direct sharing of resumes through various platforms like email, LinkedIn, or WhatsApp.
Dark Mode Support	Introduces a dark mode UI theme for a better user experience, especially in low-light environments.
Web Version	Extends the application to a web platform using Flutter Web, enabling desktop usage.

CHAPTER-9

CONCLUSION

The Resume Builder Application successfully demonstrates the application of Flutter in building cross-platform, user-friendly software for real-world needs. By combining structured form inputs, validation, dynamic template rendering, and PDF export capabilities, the application reduces the effort required to build professional resumes.

The project also highlights the importance of good software design practices such as modular architecture, reusable components, and consistent coding standards. Through this project, a deeper understanding was gained in:

- Flutter state management,
- UI/UX design principles,
- Data modeling using Singleton patterns,
- Cross-platform deployment,
- and practical software testing methods.

This application has laid the foundation for further research and development into more advanced resume-building tools. Future enhancements such as AI-powered resume feedback, cloud sync, and template marketplace can transform this into a fully-fledged professional tool for global users.

CHAPTER-10

REFERENCE

- **Canva Resume Builder:** https://www.canva.com/create/resumes/
- **Novorésumé:** https://novoresume.com/
- **Zety Resume Builder:** https://zety.com/resume-builder
- **Resume Genius:** https://resumegenius.com/resume-builder
- **Flutter Documentation:** https://docs.flutter.dev
- **Dart Programming Language:** https://dart.dev
- **Material Design Guidelines:** https://m3.material.io/
- **Google Fonts:** https://fonts.google.com (used for resume templates)
- **pdf (Resume Export as PDF):** https://pub.dev/packages/pdf
- **Local Storage:** https://pub.dev/packages/shared_preferences