INTELIDEA

Final Year Project

Session 2018-2022

A project submitted in partial fulfillment of the degree of

BS in Computer Science

Department of Computer Science

Faculty of Computer Science & Information Technology

Superior University, Lahore

FALL 2022

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R**&**D**
Area of specialization	
FYP ID	FYP-BCSM-F21-083

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-S16-027	MUHAMMAD SAIM	saaim62@gmail.com	
(ii)	BCSM-F18-235	M ARISH KHAN	arishkhan966@gmail.com	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I Muhammad Saim son of Asif Mahmood, group leader of FYP under registration no FYP-BCSM-F21-083 at Computer Science Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: Name of Group Leader: Muhammad Saim Signature: _____

Name of Supervisor: Arshia Naeem

Designation: Lecturer Signature: _____

HoD: Dr. Arfan Jaffar

Signature: _____

Project Report

INTELIDEA
(INTELIDEA)

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

PROJECT MANAGER

Comments: _____

_____

Date:_____ Signature:_____

HEAD OF THE DEPARTMENT

Comments: _____

_____

Date:_____ Signature:_____

Dedication

We are dedicating this FYP to our genius and determine coach Arshia Naeem and to our beloved parents who have meant and continue to mean so much to us. They are the beacon of hope and light to us, their support continues to regulate our lives. First and foremost, to our supervisor Arshia Naeem whose dedication and determination for research and advance projects have no bounds and, who helped us in achieving the core values of our FYP through determination and hard work. Thank you so much, we will never forget you and your efforts. Next, to our parents who supported us in the difficult times, we could not have reach so far without your help and support. Thanks to you all for supporting and believing in us

Acknowledgements

First and foremost, we really thankful to my supervisor for helping us throughout in our final year project. We are forever indebted to Arshia Naeem for her unwavering support, encouragements and patience through this process. We can never pay you back for all the help you have provided us, the experience you have helped us to gain by working for you at FYP and the precious time you spent making sure our FYP is always on track. We hope you find some kind of satisfaction in this modest paper. Thank you so much! We are very fortunate and grateful to you for providing readership and ideas; paramount to the realization of this FYP. Thank you very much for the support and helpful suggestions, we will be forever thankful to you. To Arshia Naeem, we don't know where we would be now if it wasn't for your huge help in editing our many mistakes. You are truly an outstanding person and an able educator and, we are really thankful to you from the bottom of our hearts. Not least of all, we owe so much to our whole family for their undying support, their unwavering belief that we can achieve so much. Unfortunately, we cannot thank everyone by name because it would take a lifetime but, we just want you all to know that you count so much. Had it not been for all your prayers and benedictions; were it not for your sincere love and help, we would never have completed this FYP. So, thank you all.

Executive Summary

Intelidea is an android application base platform where you can find innovative, creative, fascinating and capitalistic ideas. It is a place where Idea generators can post their ideas to get investors. And investors can easily scroll the list and choose the idea they want to invest in. With this application investors and idea generators both can greatly save their time and effort. In today's era time is money and putting your efforts in a right way to achieve the desired goals is important. Same goes for an idea generator and investor. Some of the features are mentioned as below:

Application:

- Application functionality defined as below:
- Login or Registration screen will appear.
- A pop up will be displayed on screen verifying while registering for the first time. Asking following question:
- Are you an investor or an idea generator?
- After choosing the desired option will be directed towards the required department or field.
- Note: No account remake over the platform.

Database:

Firebase Realtime database will be used that lets you store and sync-data in real-time.

OR

MongoDB which is a free and open-source cross-platform document-oriented database program. Termed as a NoSQL database program, MongoDB uses JSON-like documents with schemas.

Retrofit (JSON API):

Retrofit will be used to retrieve and upload data from database. Retrofit is a REST Client for Java and Android. It makes it relatively easy to retrieve and upload JSON (or other structured data) via a REST based webservice.

Idea Generator:

A key actor in the system that will present ideas on the platform. Idea generator is responsible for the:

- insertion of the data (ideas).
- updating of the data (ideas).
- deletion of the data (ideas).
- Connecting with investors.
- Giving access to investor over the ideas.
- Can restrict or limit access to investor over the ideas.
- Complain regarding investor behavior.
- As an idea generator only, those ideas will be displayed to the idea generator that are generated by the idea generator over the platform.

Admin:

Admin is also a key actor who will allow access to both investor and idea generator. An admin has the authority over both investor and idea generator. Some of the features are as below:

- An admin can investigate the complain of idea generator over investor authenticity of investor.
- Can directly access investor and idea generator accounts.
- Can add investor in grey list over the violation of app rules.
- Can disable or exterminate account of investor and idea generator over suspicious actions and fake accounts.

Investor:

A key actor that will view the ideas given on the platform and will interact with the idea generator. Some of the features are as below:

- Viewing Ideas.
- Connecting with Idea generators.
- Communicating with Idea generators.
- Only Ideas names will be appeared to investor.
- Due to limited access as an investor, you can only view ideas names
- After connecting with an idea generator, you can get access of ideas allowed by the idea generator.

Table of Contents

Dedication v
Acknowledgements vi
Executive Summary vii
Table of Contents x
List of Figures ..... **Error! Bookmark not defined.**
List of Tables ..... **Error! Bookmark not defined.**
Chapter 1 1
Introduction 1
1.1. Background 2
1.2. Motivations and Challenges 2
1.3. Goals and Objectives 3
1.4. Literature Review/Existing Solutions 3
1.5. Gap Analysis 4
1.6. Proposed Solution 4
1.7. Project Plan 4
1.7.1. Work Breakdown Structure 5
1.7.2. Roles & Responsibility Matrix 5
1.7.3. Gantt Chart 6
1.8. Report Outline ..... **Error! Bookmark not defined.**
Chapter 2 7
Software Requirement Specifications 7
2.1. Introduction 8
2.1.1. Purpose 8
2.1.2. Document Conventions 8
2.1.3. Intended Audience and Reading Suggestions 9
2.1.4. Product Scope 9
2.1.5. References ..... **Error! Bookmark not defined.**
2.2. Overall Description 9
2.2.1. Product Perspective 9
2.2.2. Product Functions 10
2.2.3. User Classes and Characteristics 10
2.2.4. Operating Environment 11
2.2.5. Design and Implementation Constraints 11
2.2.6. User Documentation 12
2.2.7. Assumptions and Dependencies 12
2.3. External Interface Requirements 12
2.3.1. User Interfaces 12
2.3.2. Hardware Interfaces 12
2.3.3. Software Interfaces 13
2.3.4. Communications Interfaces 14
2.4. System Features 14
2.4.1. System Feature 1 14

2.4.1.1. Description and Priority 14
2.4.1.2. Stimulus/Response Sequences 14
2.4.1.3. Functional Requirements 15
2.4.2. System Feature 2 .... **Error! Bookmark not defined.**
2.4.2.1. Description and Priority 15
2.4.2.2. Stimulus/Response Sequences 15
2.4.2.3. Functional Requirements 15
2.4.3. System Feature 3 (and so on) 15
2.5. Other Nonfunctional Requirements 17
2.5.1. Performance Requirements 17
2.5.2. Safety Requirements 17
2.5.3. Security Requirements 17
2.5.4. Software Quality Attributes 18
2.5.5. Business Rules 18
2.6. Other Requirements .... **Error! Bookmark not defined.**
Chapter 3 19
Use Case Analysis 19
3.1. Use Case Model 20
3.2. Use Case Descriptions 21
Chapter 4 26
System Design 26
4.1. Architecture Diagram 28
4.2. Domain Model 28
4.3. Entity Relationship Diagram with data dictionary 29
4.4. Class Diagram 30
4.5. Sequence / Collaboration Diagram 31
4.6. Operation contracts .... **Error! Bookmark not defined.**
4.7. Activity Diagram 32
4.8. State Transition Diagram .... **Error! Bookmark not defined.**
4.9. Component Diagram 33
4.10. Deployment Diagram 34
4.11. Data Flow diagram [only if structured approach is used - Level 0 and 1] .... **Error! Bookmark not defined.**
Chapter 5 35
Implementation 35
5.1. Important Flow Control/Pseudo codes .... **Error! Bookmark not defined.**
5.2. Components, Libraries, Web Services and stubs .... **Error! Bookmark not defined.**
5.3. Deployment Environment 36
5.4. Tools and Techniques 36
5.5. Best Practices / Coding Standards 38
5.6. Version Control 38
Chapter 6 40
Testing and Evaluation 40
6.1. Use Case Testing 41

6.2. Equivalence partitioning 41
6.3. Boundary value analysis 42
6.4. Data flow testing 42
6.5. Unit testing 42
6.6. Integration testing 43
6.7. Performance testing 43
6.8. Stress Testing 44
Chapter 7 45
Summary, Conclusion and Future Enhancements 45
7.1. Project Summary 46
7.2. Achievements and Improvements 46
7.3. Critical Review .... **Error! Bookmark not defined.**
7.4. Lessons Learnt 46
7.5. Future Enhancements/Recommendations 47
Appendices 48
Appendix A: User Manual 49
Appendix B: Administrator Manual .... **Error! Bookmark not defined.**
Appendix C: Information / Promotional Material 50
Reference and Bibliography 54
Index 56

List of Figures

1.1	Caption of first figure of first chapter	6
1.2	Caption of second figure of first chapter	7
2.1	Caption of first figure of second chapter	14
2.2	Caption of second figure of second chapter	22
2.3	Caption of third figure of second chapter	26
5.1	Caption of first figure of fifth chapter	49
5.2	Caption of second figure of fifth chapter	49

List of Tables

1.1	label of first table of first chapter	6
1.2	label of second table of first chapter	7
2.1	label of first table of second chapter	14
2.2	label of second table of second chapter	22
2.3	label of third table of second chapter	26
5.1	label of first table of fifth chapter	49
5.2	label of second table of fifth chapter	49

Chapter 1

Introduction

Chapter 1: Introduction

In this world that we are living is flourishing with state-of-the-art technology and ideas. Every single day trillions of ideas are provoked in billions of minds. And millions of entrepreneurial minds are finding ways to put their investments at safe and productive projects. Intelidea is an android app based on the concept of an innovative and hybrid platform that is providing an opportunity for the investors and idea generators to interact and communicate with each other efficiently and ease. It is first of its kind. There has been no other platform like this before has been made where an investor and idea generator can interact and communicate easily. Bringing investors and idea generators at one place.

1.1. Background

The investment view can be really dynamic and ever-evolving. However, individuals who take the time to understand the fundamental ideas and the many asset classes stand to benefit greatly in the long run. Learning how to differentiate between various investment types and what each one's "risk-level" is the first step. Investing can be a frightening proposition for beginners, with a huge variety of potential assets to add to a portfolio. Those thinking of investing via an investment platform will certainly have questions regarding their key functions. However, the investment platform market study directed by the FCA notes that even those who are presently investing via a platform are still uncertain of the details of how their investment works. An investment platform is basically an online service which lets you to buy, sell and hold funds. It’s likely for you to do this yourself directly on a non-directed basis via a D2C (direct to customer) platform, or on a recommended basis using a financial consultant who will invest on your behalf.

1.2. Motivations and Challenges

Over the years, platforms have gained a lot of popularity since they can be a tax-efficient way to hold investments. It’s viable to gain access to a vast fund range, which can occasionally be

accessible at a discounted rate - although, it's vital to note that the funds offered can differ between providers. Investors also have the ease of dealing with multiple products under one roof as many platforms allow you to hold an ISA, a GIA and a SIPP concurrently. Having online admission also allows you to view and check the progress of your investments at any time. Depending on the type of platform service you choose, this could mean that probable adjustments could quickly be made in line with market changes.

1.3. Goals and Objectives

Objectives

To provide ground breaking and innovative ideas to investors and entrepreneurial mindsets and providing a platform for idea generators to present their efficient and innovative ideas to the world.

Goals

Providing investor and idea generator a single platform to find better ideas and upload ideas and fund them easily and effortlessly. Rather than struggling in industries or corporations.

1.4. Literature Review/Existing Solutions

Intelidea is a new idea and till now we cannot find any existing platform like this where investors and idea generators can virtually meet at the same platform. That's why we feel the need for this platform and that's why we are creating this application. In the last five years, the propagation of subsidizing platforms has certainly provided new prospects for entrepreneurs. Research recommends that in 2017, there are more than 2000 active platforms worldwide (Galkiewicz and Galkiewicz, 2018). Moreover, the outbreak of COVID-19 has seen the subsidizing market witnessing significant growth (Mordo intelligence 2020). However, success rates among subsidizing platforms remain fairly low. Industry assisted research by Mollick (2014) based on a database from Kickstarter reported a success rate between 2009 and 2012 of 48.1 percent. Data for 2015 and 2016 specify that success rates have actually declines to 36 percent and 32 percent respectively. Likewise, data from Indiegogo, the second largest subsidizing platform, reported success rates to be about 33 percent (Massolution, 2015).

Perhaps, the advance in the number of platforms and the failure rates among drives are organized, since the growth in subsidizing platforms while beneficial, may cause the attention of potential investors to be spread too thin.

1.5. Gap Analysis

It is the comparison of real performance with desired performance; that is the existing state and the desired future state. An important feature of gap analysis is finding what needs to be done in a Project.

1.6. Proposed Solution

In this proposal we are proposing a platform where investors and idea generators will be linked together. Some of the main features are as follows:

- **Time management:**

  As an idea generator you have to explore investors manually and while exploring investors a lot of your time gets wasted. But with our app your efforts and time for finding ideas are just a matter of a few clicks away.

- **Close Interaction:**

  With the help of this platform investors and idea generators are linked. No third person involvement and lengthy procedures for decision making over the ideas and products. They can decide the meeting methods, investment methods and legalization of investment

1.7. Project Plan

We will start with the UI screens prototype designing so that we can easily figure out the features and functions later. After completing the UI prototype, we will start working on the online database. Then we will start working on the code and development. We will implement the MVVM architecture and then network layer. Then we will start the QA testing and then finally generate the final APK build for users.

1.7.1. Work Breakdown Structure

The work breakdown structure will be as follows:

- Prototype designing.
- Online database setup.
- JSON API creation.
- Screen designing in app.
- MVVM architecture implementation.
- Koin dependency injection implementation.
- Network layer implementation.
- Code testing.
- Quality assurance testing.
- Bugs fixing.
- Quality assurance cycle 2 testing.
- Generate the Apk build.

1.7.2. Roles & Responsibility Matrix

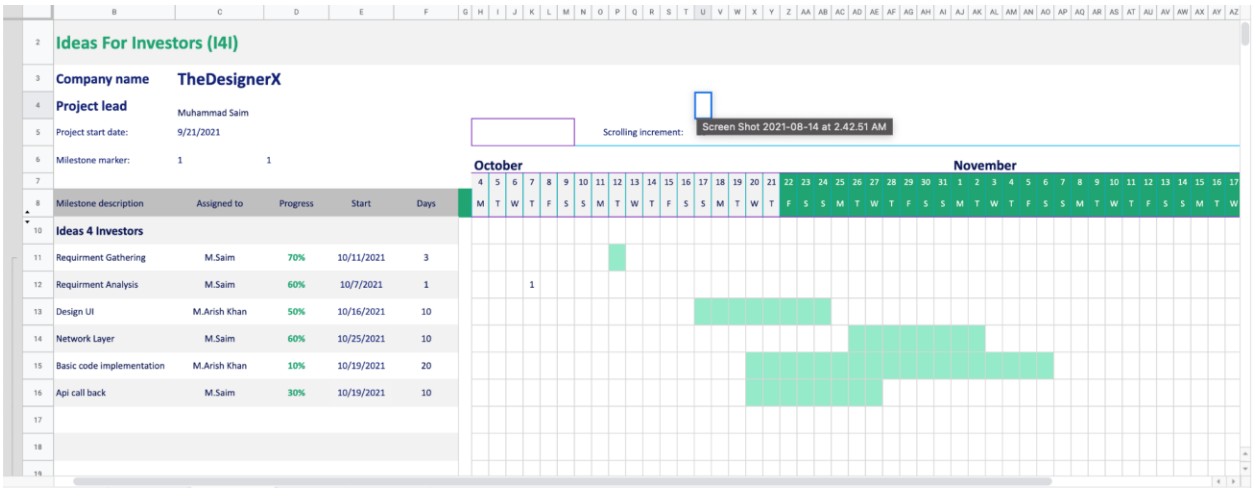
The purpose of roles & responsibility matrix is to identify who will do what.

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Requirement analysis	1	Done	2	M.Saim
2	Prototype Design	2	Done	5	M.Arish Khan
3	UI design	3	Done	3	M.Arish Khan
4	MVVM	4	Done	2	M.Saim
5	Koin injection	5	Done	1	M.Saim
6	Network layer	6	Done	2	M.Saim
7	Code testing	7	Done	1	M.Saim
8	QA testing	8	Done	3	M.Saim and

					M.Arish Khan
9	Bugs fixing	9	Done	1	M.Arish Khan
10	QA cycle 2	10	Done	1	M.Saim and M.Arish Khan
11	APK build generation	11	Done	1	M.Saim

1.7.3. Gantt Chart

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]



Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

2.1.1. Purpose

What intelidea is designed to do is providing a platform or place where an investor and idea generators can communicate with each other remotely regardless of arranging meetings at different places. The scope of intelidea is global. Providing efficient and innovative solution effortlessly. Global Scope

2.1.2. Document Conventions

This document uses the following conventions

Term	Description
User	Somebody who cooperates with the mobile application.
Admin/Administrator	System administrator who is given precise permission for managing and operating the system.
Investor	The person who will put his funds on an idea or project to achieve desired goals.
Idea Generator	The person who gives ideas over the platform and get funds from the investor.
GPS	Global Positioning System
Play Store	An application on mobile that helps user to find new suitable applications with mobile platform and download them from Internet.
Stakeholder	Any person who has collaboration with the system who is not a developer.

JSON	Javascript Object Notation
Retrofit	REST Client library for getting response.
MVVM	Model-View-View-Model used for holding app data.
DB	Database
ER	Entity Relation Diagram

2.1.3. Intended Audience and Reading Suggestions

This project is a model for the investors and idea generators and it has a global scope. This project is useful for the investors and idea generators.

Investor and idea generator. Providing a platform for investors to invest on innovative ideas and idea generators to present their innovative ideas and get them funded efficiently and effortlessly.

2.1.4. Product Scope

The motive of Intelidea is to ease investment procedure and create a suitable and easy-to-use application for investors, trying to put their investments over innovative and efficient ideas. And Idea generators to present their ground breaking and innovative ideas to the investors. The system is based on an android app. Providing a platform to the community of idea generators and investors. Above all, we hope to provide a contented user experience.

2.2. Overall Description

2.2.1. Product Perspective

An android mobile app that will stores the following information.

Investor Details:

It includes information regarding investors and their objectives. The types of projects in they are interested to invest on. Investor whereabouts, past experience, profession and etc.

Idea Generator Details:

It includes information regarding idea generators and their objectives. Category of ideas they are presenting. Idea generator personal information.

Investments and Ideas description:

Details about the investments and ideas that are getting funded by different investors. Details about different ideas. What that particular idea is for and long term uses. Cost required for biding or funding that idea.

2.2.2. Product Functions

- **Time management:**

  An idea generator has to explore investors manually and while exploring investors a lot of time is wasted. But with Intelidea the efforts and time for finding ideas are just a matter of few clicks away.

- **Close Interaction:**

  With the help of this platform investors and idea generators are linked. No third person involvement and lengthy procedures for decision making over the ideas and products. They can decide the meeting methods, investment methods and legalization of investment

2.2.3. User Classes and Characteristics

Idea generators will be able to share their ideas and these ideas will be stored in a live server-side database for efficient retrieval of data. Investors will be able to access ideas list once all of the required credentials regarding investors identity and information are verified. The app will aid three types of user rights, Investor, idea generator and Admin. Investor will have access to investor functions, and the idea generator will have the access to upload ideas only. Both of them can communicate with each other. Admin has a wide range of access over the app for maintaining app's security and reliability. Some of the functions regarding investor are as follow:

Search for Ideas:

- Category
- Computer relative
- Mechanics relative.

- Tech relative.
- Biding an idea.

Idea generator should have following functionalities:

- Create an idea.
- Search for investors.
- Communicate with investor.
- Report any suspicious investor.

Admin have following functionalities:

- Access over ideas
- Delete any idea
- Kick any suspicious or malicious investor or idea generator.
- Maintain app work flow.

2.2.4. Operating Environment

Operating environment for the Intelidea is as listed below.

- Live Database
- client/server system
- Operating system: Windows.
- Database: MongoDB, Firebase
- Platform: Java/Kotlin
- Google Map Api.
- Android Studio.
- Rx Java.

2.2.5. Design and Implementation Constraints

After all the major issues that intelidea is facing is the authenticity of investor and security issues regarding investment. Whether an investor is a valid or authentic investor is an investor determine to fund the idea or just going to steal that idea. The Internet linking is also a

restriction for the application. Since the application fetches data from the database over the Internet, it is critical that there is an Internet connection for the application to function.

2.2.6. User Documentation

User Documentation is the part in which you describe how you're going to distribute your documentation to your customer and product users. So right now, our current manual or guidelines to operate this app is our document.

2.2.7. Assumptions and Dependencies

One supposition about the product is that it will always be used on mobile phones that have sufficient performance. If the phone does not have enough hardware resources available for the application, for example the users might have placed them with other applications, there may be scenarios where the application does not work as proposed or even at all.

2.3. External Interface Requirements

2.3.1. User Interfaces

Prototype designing

- Adobe XD.

Frontend Layouts/Tools:

- Constraint layout.
- Linear layout.
- CardView.

Backend:

- MongoDB
- Firebase

2.3.2. Hardware Interfaces

For development, the hardware requirements are the following:

- Desktop / Laptop with 8GB RAM and 64-bit operating system

- High speed online servers.
- Good internet connection.
- Multiple android devices with at least 4GB RAM and 128 storage and android 9.1 pie version.

2.3.3. Software Interfaces

Following are the software interfaces that will be used for Intelidea.

Software	Description
Operating System	Choosing Windows operating system for its best support and user-friendliness.
Platform	Android
Programming Languages	Java / Kotlin for developing app that will run on server side.
Database	MongoDB / Firebase is used because of modern post-relational databases that allow for flexibility, speed and storing and retrieving big data and high-performance use cases.
Editor	Android Studio as the official android app IDE. Designed to build high quality apps for every android device.
Version Control System	Source tree for enabling data availability and access.
Coding Architecture	MVVM for separating Data presentation logic from core business logic part.

2.3.4. Communications Interfaces

Intelidea supports all types of android devices. The communication between the dissimilar parts of the system is vital since they depend on each other. However, in what way the communication is accomplished is not important for the system and is therefore controlled by the primary operating system for the app.

2.4. System Features

Functional requirements are product qualities or functions developers must implement to allow users to accomplish their tasks. So, it's significant to make them both clear for the development team and the stakeholders. Usually, functional requirements define system behavior under explicit conditions. It defines a software system or its component. A function is nothing more than the inputs, behavior, and outputs of the software system. It can be a computation, data handling, business process, user collaboration, or any other precise functionality which describes what function a system is likely to perform. Following are the basic functions and actions that the system will be able to perform

2.4.1. Create New Idea

2.4.1.1. Description and Priority

Description: The system shall create an idea in the database.
Priority: High

2.4.1.2. Stimulus/Response Sequences

- Pre-condition: The system shall create an investor request to the database to access ideas.
- Post-condition: The investor will get the idea.
- Normal flow: The investor shall first search for the ideas from the idealist if idea found then display get an idea panel button which on click shall fetch the idea from the database.
- Alternative flow: A pop up message is displayed in case of idea of particular idea not found.

- Special condition: Idea-ID must be available in case of idea creation.

2.4.1.3. Functional Requirements

FR-1: Create an idea

FR-2: idea will be upload in idea list

FR3-3: investor can access idea

2.4.2. Create Investor Request

2.4.2.1. Description and Priority

Description: The system shall create an investor request to the database to access ideas.

Priority: Low

2.4.2.2. Stimulus/Response Sequences

- Pre-condition: The system had created a request from admin to allow access to the idea list fetched from the database.
- Post-condition: The investor will get the desired idea displayed on the screen.
- Normal flow: To access the ideas from investor side investor has to send a request through access interface to admin.
- Alternative flow: A pop up message is displayed in case of idea of particular idea not found.
- Special condition: Investor must be available in case of request.

2.4.2.3. Functional Requirements

FR-1: Create an investor request

FR-2: idea will be upload in idea list.

FR3-3: investor can access idea.

2.4.3. Search Ideas

2.4.3.1. Description and Priority

Description: The investor will search ideas from the given idea list.

Priority: High.

2.4.3.2. Stimulus/Response Sequences

- Pre-condition: The investor search for the ideas using specific Idea-ID allotted with each idea-on-idea list.
- Post-condition: The investor will send a request
- Normal flow: To access the ideas from investor side investor has to send a request through access interface to admin.
- Alternative flow: To access the ideas from investor side investor has to send a request through access interface to admin.
- Special condition: Idea-ID must be available in case of searching ideas from the idea list.

2.4.3.3. Functional Requirements

FR-1: Search Ideas.

FR-2: idea will be upload in idea-list.

FR3-3: investor can access idea.

2.4.4. Contact Idea Generator

2.4.4.1. Description and Priority

Description: The investor will contact the idea generator after choosing the desired idea.

Priority: High.

2.4.4.2. Stimulus/Response Sequences

- Pre-condition: The investor have to choose idea from the idea list first.
- Post-condition: The idea generator will be able to contact idea generator.
- Normal flow: The idea generator will be able to contact idea generator.
- Alternative flow: To access the ideas from investor side investor has to send a request through access interface to admin.

2.4.4.3. Functional Requirements

FR-1: Contact Idea Generator.

FR-2: Deal with idea generator over the idea.

2.5. Other Nonfunctional Requirements

These requirements are more focused because whole system depends on its nonfunctional requirements. Some are below:

2.5.1. Performance Requirements

Prominent search feature:

- TITLE: Prominent search feature
- DESC: The search feature should be prominent and easy to find for the user.
- RAT: In order to for a user to find the search feature easily.
- DEP: none

Category feature:

- TITLE: Category feature
- DESC: The different types of categories available and should be evident, simple and easy to understand.
- DEP: none

2.5.2. Safety Requirements

If there is an extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

2.5.3. Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

2.5.4. Software Quality Attributes

AVAILABILITY

24/7 availability of thousands of ideas of different categories over the app.

RELIABILITY

The system shall generate no result found upon invalid input by user.

MAINTAINABILITY

The administrators will maintain the app and user check and balance. The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.

Flexibility

The system would contain loosely coupled modules, which make the system flexible enough to make changes and facile to update. Furthermore, the flexibility of the system makes it easy to maintain. This allows the changes to be quicker and the system easily adapts to the new design.

REUSABILITY

There are a number of components that will be developed and integrated together in order to completely develop the system. These modules that would be developed, have their own individual functionality that can be used later for other purposes.

2.5.5. Business Rules

A business rule is a constraint of the business itself that may guide system development. It is a rule that must be followed, no matter what else is happening. It often involves very specific criteria or conditions for compliance. All users need a valid email address.

Business rules can be classified in several ways, and they can vary in their classification depending on the source of information. However, irrespective of their categorization business rules are typically expressed using formal logic qualifiers, such as: "IF-THEN", "IF-ELSE", "ONLY IF", "WHEN", et cetera. This syntax is used across the following different types of business rules:

- **Constraint rules**
- **Derivation rules**

Chapter 3

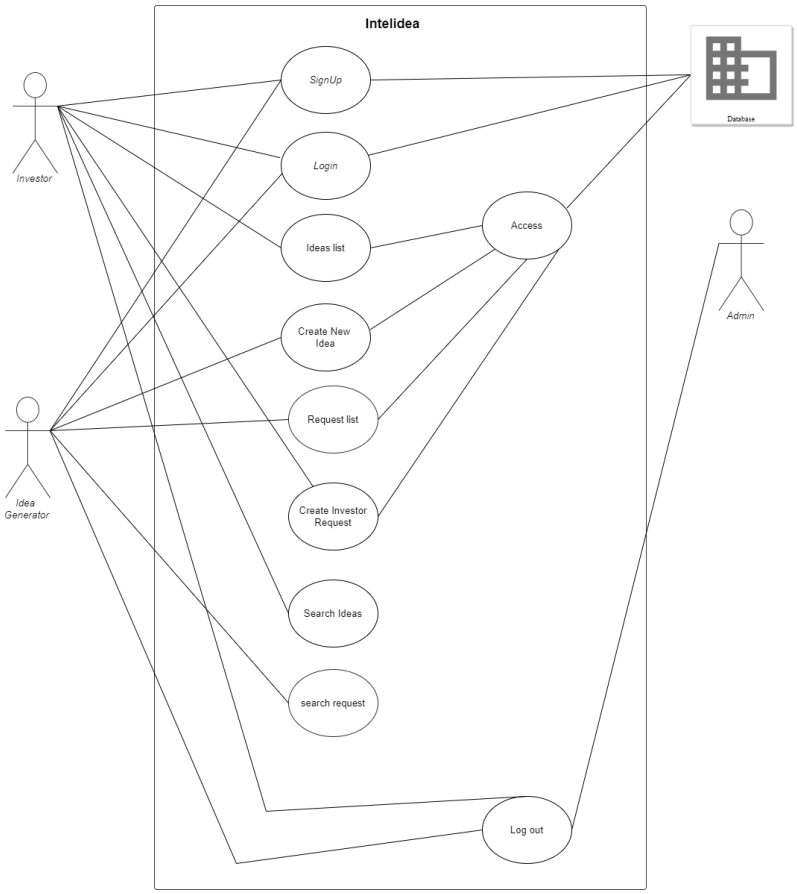
Use Case Analysis

Chapter 3: System Analysis

System analysis is done to investigate a system or its components in order to pinpoint its goals. It is a problem-solving method that improves the system and guarantees that all the components of the system work proficiently to accomplish their purpose. Analysis specifies what the system should do.

3.1. Use Case Model

Following is the use case diagram which depicts the system features exposed to the user.



Use Case Model Diagram

3.2. Use Case Descriptions

Use case descriptions provide complete description and details of the procedures used to perform the system functions.

Use Case 01

Create New Idea	
ID:	UC-1
Use case:	Create New Idea
Actor:	Idea Generator
Type:	Primary
Description:	The system shall create an idea in the database.
Pre-condition:	The system had created an idea and stored it into the database.
Post-condition:	The investor will get the idea.
Normal flow:	The investor shall first search for the ideas from the idealist if idea found then display get an idea panel button which on click shall fetch the idea from the database.
Alternative flow:	A pop up message is displayed in case of idea of particular idea not found.
Special condition:	Idea-ID must be available in case of idea creation.
Assumption:	Access to the database and internet.
Frequency of use:	High.

Use Case 02

Create investor request	
ID:	UC-2
Use case:	Create Investor Request
Actor:	Investor
Type:	Primary
Description:	The system shall create an investor request to the database to access ideas.
Pre-condition:	The system had created a request from admin to allow access to the idea list fetched from the database.
Post-condition:	The investor will send a request
Normal flow:	To access the ideas from investor side investor has to send a request through access interface to admin.
Special condition:	Investor must be available in case of request.
Assumption:	Access to the admin and internet.
Frequency of use:	Low.

Use Case 03

Search Ideas	
ID:	UC-3
Use case:	Search Ideas
Actor:	Investor
Type:	Primary
Description:	The investor will search ideas from the given idea list.
Pre-condition:	The investor search for the ideas using specific Idea-ID allotted with each idea-on-idea list.
Post-condition:	The investor will get the desired idea displayed on the screen.
Normal flow:	To access the ideas from investor side investor has to send a request through access interface to admin.
Special condition:	Idea-ID must be available in case of searching ideas from the idea list.
Assumption:	Access to Idea-ID and internet.
Frequency of use:	High.

Use Case 04

Show single idea details	
ID:	UC-4
Use case:	Show single idea details
Actor:	Idea Generator
Type:	Primary
Description:	The idea generator will be displayed only those ideas which are generated by them.
Pre-condition:	The idea generator can access his idea history through the given history of idea option.
Post-condition:	The idea generator will get the history of ideas uploaded by him displayed.
Normal flow:	The idea generator can access his idea history through the given history of idea option.
Assumption:	Idea-ID and internet.
Frequency of use:	low.

Use Case 05

Contact Idea Generator	
ID:	UC-5
Use case:	Contact Idea Generator
Actor:	Investor
Type:	Primary
Description:	The investor will contact the idea generator after choosing the desired idea.
Pre-condition:	The investor have to choose idea from the idea list first.
Post-condition:	The idea generator will be able to contact idea generator.
Normal flow:	The idea generator will be able to contact idea generator.
Assumption:	Idea generator availability and internet.
Frequency of use:	High.

Chapter 4

System Design

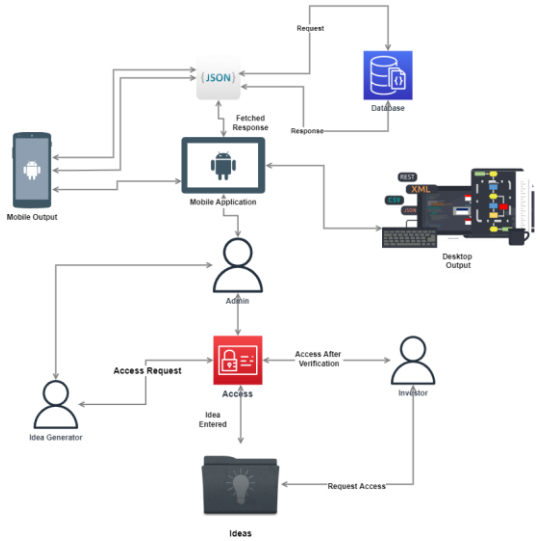
Chapter 4: System Design

The process of creating a system's components, including its architecture, modules, and components, as well as its many interfaces and the data it processes, is known as system design. A good system design will arrange the program modules so that they are simple to modify. Developers can manage the size and complexity of program with the aid of structured design strategies. For developers, analysts write instructions on how to write code and how to fit parts of code together to make program.

Components of a system design are as follow:

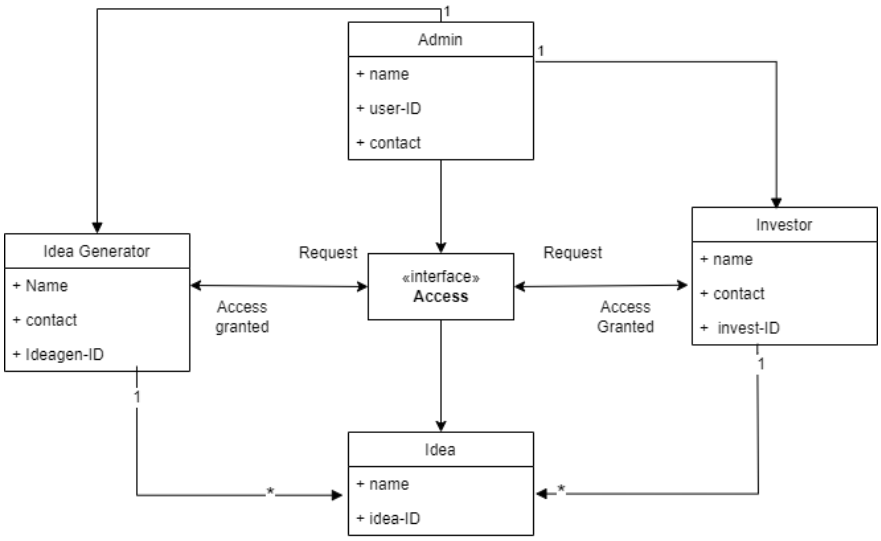
- Architecture
- Modules
- Components
- Interfaces
- Data

4.1. Architecture Diagram



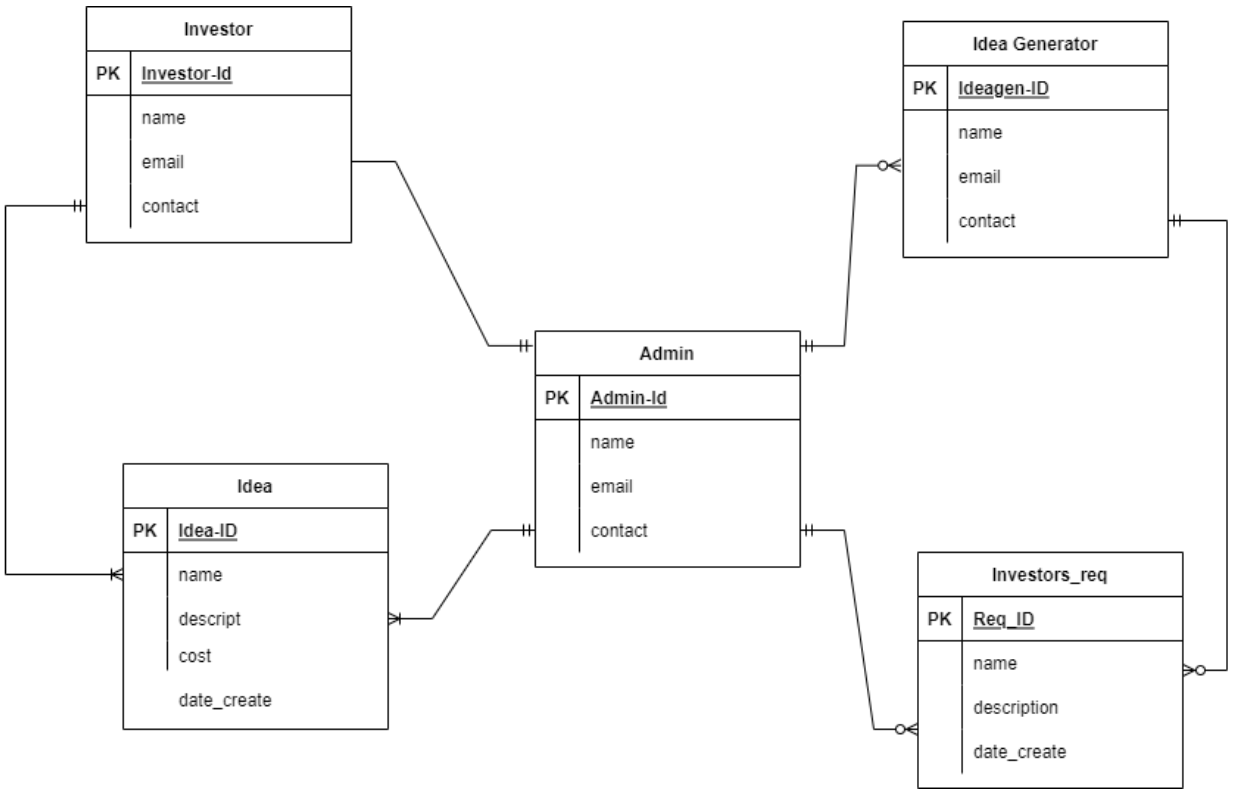
4.1 Architecture Diagram

4.2. Domain Model



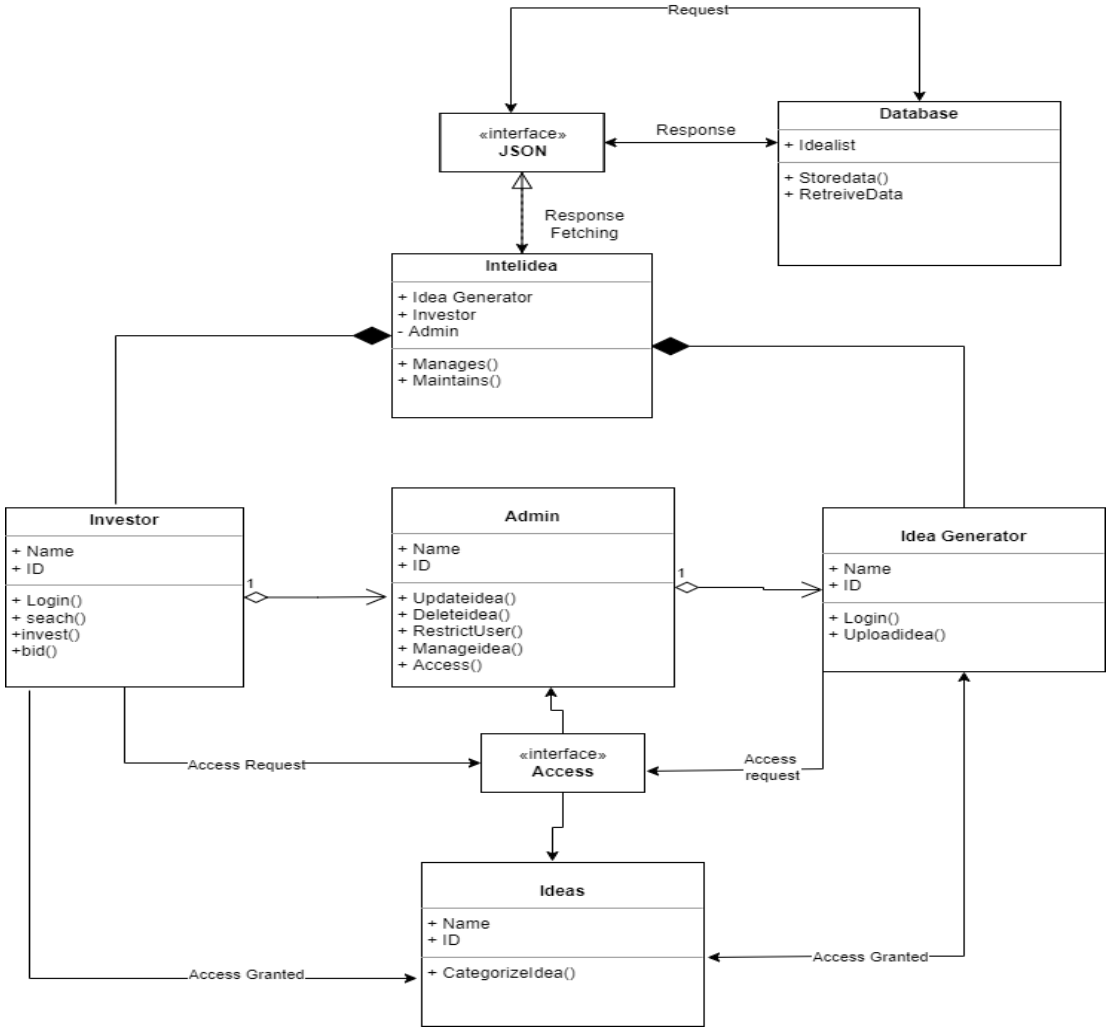
4.2 Domain Model

4.3. Entity Relationship Diagram with data dictionary



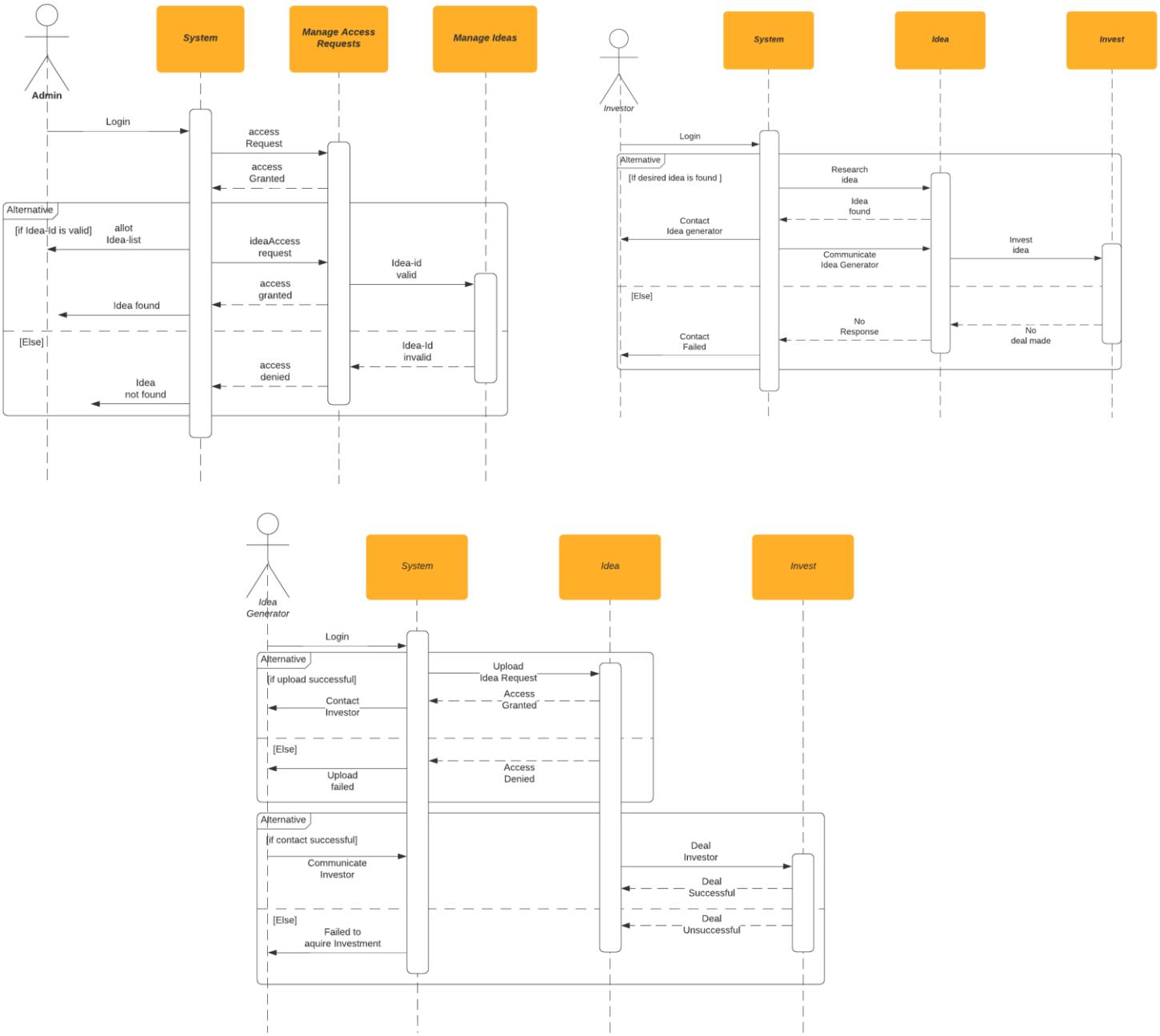
4.3 ER Model

4.4. Class Diagram



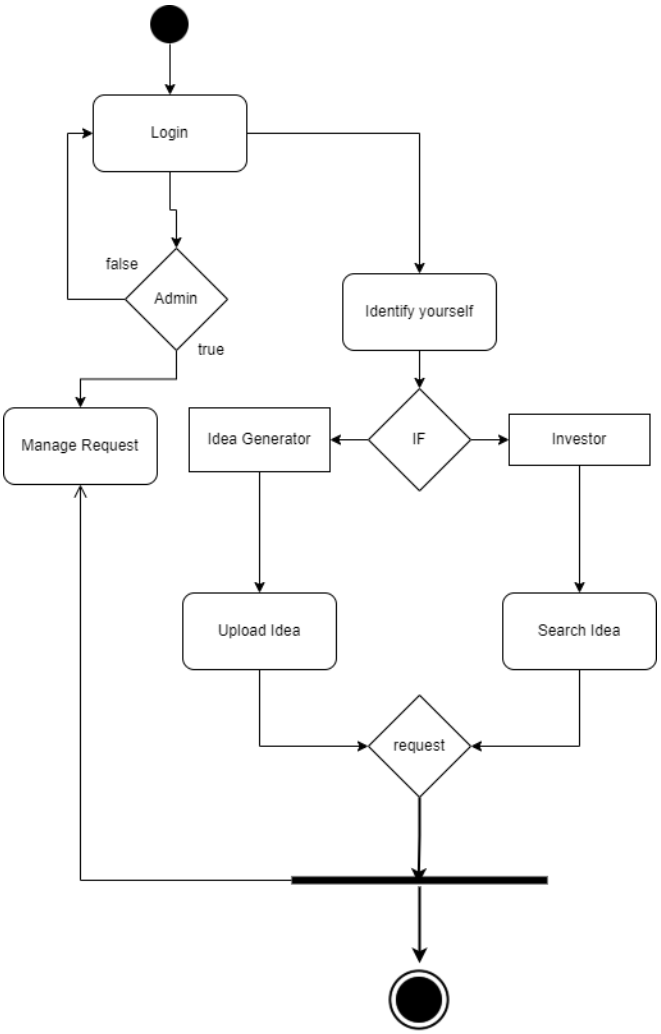
4.4 Class Diagram

4.5. Sequence / Collaboration Diagram



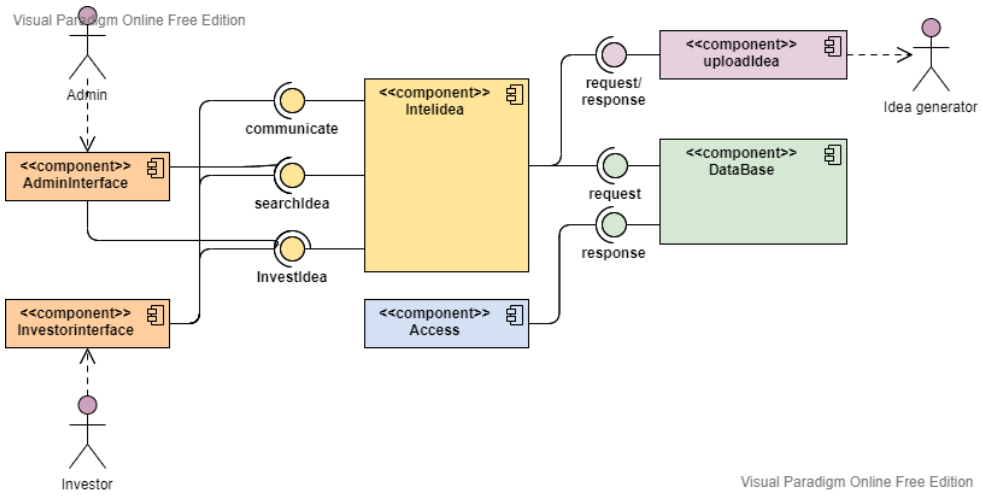
4.5 Sequence Diagram

4.6. Activity Diagram



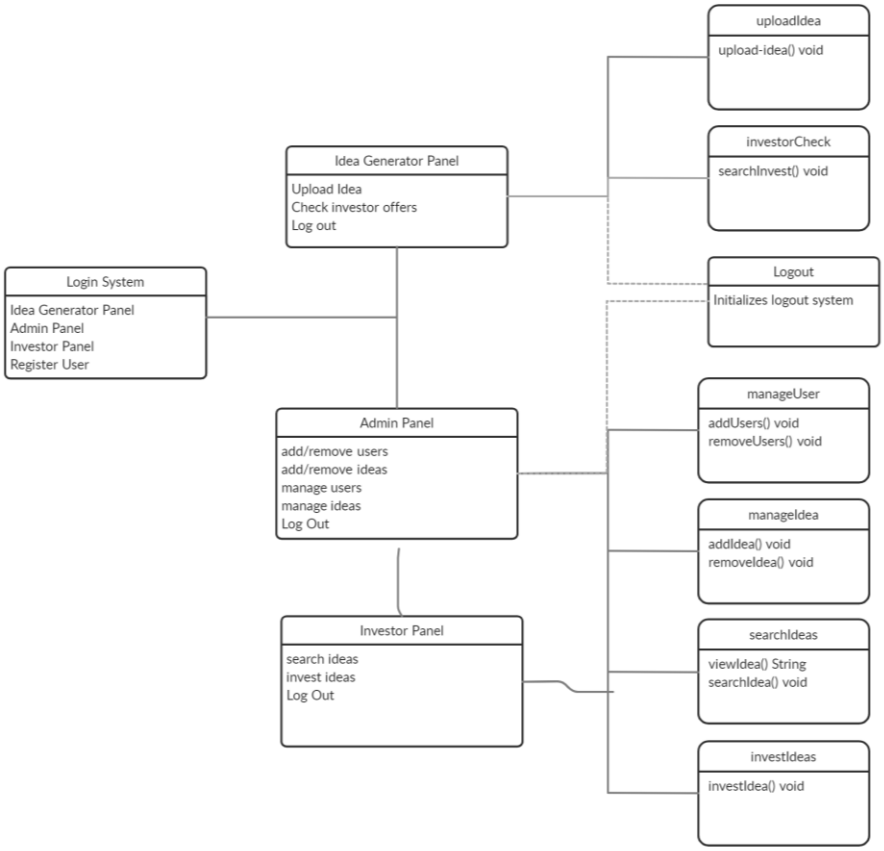
4.6 Activity Diagram

4.7. Component Diagram



4.7 Component Diagram

4.8. Deployment Diagram



4.8 Deployment Diagram

Chapter 5

Implementation

Chapter 5: Implementation

The project team really works on the project during the implementation phase to create the deliverables. Anything your project delivers is referred to as a "deliverable". All of the goods or services that the project provides to the client, customer, or sponsor are considered deliverables, as are all the project management documentation you create.

Following are the implementation steps:

- Investor module.
- Idea Generator Module.
- Biding.
- Idea Lists

5.1. Deployment Environment

IDE Android Studio

Android studio is the deployment environment that is used. The official Integrated Development Environment (IDE) for android application development. It delivers more features that enhance our productivity while building Android apps. The application has been developed using Android Studio 4.1.

- A fast and feature-rich emulator.
- A unified environment where you can develop for all Android devices.
- Apply Changes to push code and resource changes to your running app without restarting your app.

5.2. Tools and Techniques

The Below mentioned tools and techniques will help us to create this application.

Coding language

- Kotlin
- Java

Database

- MongoDb
- Firebase

Dependency injection

- Koin
- Network layer
- Retrofit 2 with Json Api's

Coding architecture

- MVVM

Frontend layouts

- Constraint layout
- Linear layout
- CardView

Other Technology

- Google Map Api
- AndroidX
- Rx Java

Version control system

- Source tree

Prototype designing

- Adobe XD

Communication tool

- Slack

Api testing

- Postman

Editor

- Android studio

5.3. Best Practices / Coding Standards

MVM Koin Kotlin

For the best practices and coding standards MVVM Koin Kotlin injection and Kotlin framework is used. Model-View-View-Model (MVVM) is a widely used software architecture pattern that disables all downsides of MVP and MVC design patterns. MVVM recommends splitting up the data presentation logic (Views or UI) from the core business logic part of the application. Koin is a frivolous dependency injection framework, that uses Kotlin's DSLs to idly resolve dependency graph at runtime. Dependency injection is a programming method that makes a class independent of its dependencies. For koin dependency injection we would need Kotlin. Kotlin is a cross-platform programming language that may be used as a substitute to java for android app development. Kotlin is a multi-platform programming language that can be executed with ease on a java virtual machine. It has following properties:

- Expressive and concise
- Safer code
- Structured concurrency
- Interoperable

5.4. Version Control

Github

For version control system Github is used. For software developers, GitHub is a web-based platform for version control and collaboration. A distributed version management system called Git keeps track of several versions of a file (or set of files). It enables users to access, contrast, update, and disseminate any recorded version whenever they want. Other framework that is used for version control is sourcetree.

Source Tree

Sourcetree helps you focus on coding by making it easier to interact with your Git repositories. Use the straightforward Git GUI from Sourcetree to visualize and manage your repositories. Perfect for increasing productivity among seasoned users. Among other things, review changesets, stash, and cherry-pick between branches. A completely functional GUI that comes pre-packaged with an effective, reliable development process use Git and Mercurial.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

Through testing, a system or component is compared to requirements and specifications as part of the test and evaluation **(T&E)** process. The outcomes are assessed in order to gauge design, performance, supportability, advancement etc. It is a methodical, impartial evaluation of an ongoing or finished project. The testing of the finished product is referred to as software testing. It examines issues like whether the app functions properly and whether all modules are up to par in terms of performance.

6.1. Use Case Testing

Using the use case testing is a method that helps to classify test cases that cover the whole system, on a transaction-by-transaction basis, from start to finish. It is a description of how a user specifically used the system. It is frequently used when creating tests or systems to determine acceptable levels. Our cases are as follows:

- Create New Idea.
- Create Investor Request.
- Search Ideas.
- Show single idea details.
- Contact Idea Generator.

Below are the main actors in our product:

- Investor.
- Idea Generator.

6.2. Equivalence partitioning

A software testing approach called equivalence partitioning or equivalence class partitioning (ECP) separates the input data of a software unit into divisions of equivalent data from which test cases can be constructed. Test cases should theoretically cover each partition at least once. The requirements and specifications of the software serve as the basis for the equivalence partitioning. The benefit of this strategy is that by going from an infinite to a finite number of

test cases, it helps to shorten the testing period. It is appropriate for use at every stage of the testing procedure.

- It divides the input data of software into different equivalence data classes.
- You can apply this technique, where there is a range in the input field.

6.3. Boundary value analysis

Testing borders or extreme endpoints between input value segments is known as boundary testing.

Therefore, boundary values are those extreme endpoints, such as Start-End, Lower-Upper, Maximum-Minimum, and Just Inside-Just Outside values, and boundary testing is the term used to describe the testing.

Normal boundary value testing's fundamental tenet is to choose input variable values at:

- Minimum.
- Just above the minimum.
- A nominal value.
- Just below the maximum.
- Maximum.

6.4. Data flow testing

A white-box testing method called data flow testing looks at the data flow in relation to the variables utilized in the code. It looks at how variables are initialized and verifies their values on each occasion. A data flow model is a diagrammatic description of how information moves through a system and is exchanged. The procedures involved in transporting data from input to file storage and report production are described in data flow models, which are used to graphically illustrate the flow of data in an information system.

6.5. Unit testing

The smallest testable components of an application, known as units, are separately and independently examined for appropriate operation as part of the unit testing phase of software

development. A type of testing called unit testing involves testing individual software modules or functions. Its main objective is to test each component or function. The smallest testable component of an application is called a unit. It typically provides a single output and only has one or a few inputs. Unit testing's primary goal is to confirm that each component is functioning properly and according to plan. Unit testing aims to:

- To isolate a section of code.
- To verify the correctness of the code.
- To test every function and procedure.
- To fix bugs early in the development cycle and to save costs.
- To help the developers to understand the code base and enable them to make changes quickly.
- To help with code reuse.

6.6. Integration testing

Integration testing is a sort of testing in which software modules are conceptually integrated and tested collectively. Multiple software modules created by various programmers make up a typical software project. This level of testing aims to identify issues with how various software modules interact when they are combined. The main goal of integration testing is to examine how well these modules communicate data. The term "integration and testing" (I&T), "string testing," and occasionally "thread testing" are also used to describe it.

6.7. Performance testing

Performance testing is a method for testing software applications' responsiveness, scalability, dependability, stability, and resource utilization under different workloads. Performance testing is mostly used to locate and fix software application performance issues. It falls under performance engineering and is additionally referred to as "Perf Testing." Performance testing is focused on examining a software program's:

- Speed – Determines whether the application responds quickly.
- Scalability – Determines maximum user load the software application can handle.

- Stability – Determines if the application is stable under varying loads.

6.8. Stress Testing

Software testing known as "stress testing" is used to confirm the dependability and stability of software applications. The purpose of stress testing is to evaluate a piece of software's robustness and error-handling abilities under conditions of extremely high load and to make sure that it won't crash under pressure. It even conducts testing that goes beyond typical operational conditions to assess how software performs in harsh environments. Endurance testing is another name for stress testing. AUTs undergo brief periods of stress as part of stress testing to determine their endurance. Finding the breaking point of a system, piece of software, or piece of hardware is one of the most common uses of stress testing. Additionally, it examines how well the system handles errors in challenging situations.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

Intelidea is an android application base platform where you can find innovative, creative, fascinating and capitalistic ideas. It is a place where Idea generators can post their ideas to get investors. And investors can easily scroll the list and choose the idea they want to invest in. With this application investors and idea generators both can greatly save their time and effort. In today's era time is money and putting your efforts in a right way to achieve the desired goals is important.

7.2. Achievements and Improvements

The main purpose of bringing an investor and idea generator at one place has been achieved through using this platform. Some achievements are mentioned below:

- Overcoming the communication barrier
- Deal with ease.
- Explore and invest on different and stunning ideas.

Improvements that will be made with time and further user experience. Right now we have introduced some improvements such as:

- Search ideas.
- Create investor request.
- Search request.

7.3. Lessons Learnt

Lessons learned are the information that has been recorded and reflects both the successful and unsuccessful aspects of a project. They stand for both the company's dedication to excellent project management and the project manager's chance to gain knowledge from others' real-world experiences. Some of them are mentioned are below:

- Demand for an efficient and powerful communication channel between individual related to their domains in our case its idea generator and investor.
- How to reduce the communication barriers.
- Ways to resolve problems that are based on modern era issues.

7.4. Future Enhancements/Recommendations

In near future Intelidea has a very broad range. In the future, the project might be carried out internationally. The project is quite expandable, so it can be upgraded in the near future as and when the need for it arises. The customer is now able to manage and, as a result, conduct the complete project in a lot better, accurate, and error-free manner thanks to the proposed software of database Space Manager being available and fully working. The following are the most prominent upgrades and improvements that are the main focus:

Security

Security is an issue that cannot be undermined. Now, these days security breaches and parameter bypass is pretty common. Currently trying to implement the security in our project. But facing consistent real-time problems that are making it tough to implement security. But in near future updates more enhancements are expected depending hugely on the user experience. We are pretty determined that security module will be prepared as soon as possible and will efficiently deal with the real time security issues.

Anti-Fraudulent

Fraud is the greatest deception and can be a headache that raises numerous problems and questions related towards the platform transparency. Anti-Fraudulent module is under consistent consideration. In near future, it would be possible to provide an anti-fraudulent update depending on user experience and reviews.

Appendices

Appendix A

A user manual is a guidebook that is given to a user to aid in the effortless use of a specific system, product, or service. It is often referred to as a user handbook or an instruction manual. These documents provide in-depth details about operations, standards and guidelines, troubleshooting manuals, features, and more. User guides typically include step-by-step guidance on how to use a product as well as potential troubleshooting in the event that something goes wrong. It should have a table of contents and index to make it easier for consumers to identify the passages that apply to their problem as it's not necessarily intended to be read from cover to cover.

Appendix A
Intelidea

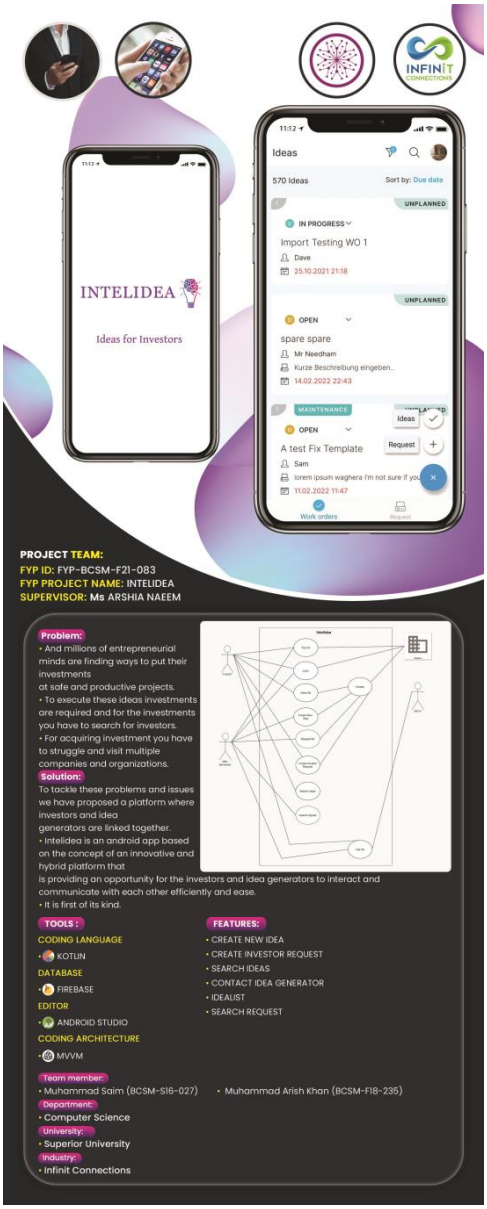
Intelidea is an android application base platform where you can find innovative, creative, fascinating and capitalistic ideas. It is a place where Idea generators can post their ideas to get investors. And investors can easily scroll the list and choose the idea they want to invest in. With this application investors and idea generators both can greatly save their time and effort. In today's era time is money and putting your efforts in a right way to achieve the desired goals is important.

Intelidea is an android application base platform where you can find innovative, creative, fascinating and capitalistic ideas. It is a place where Idea generators can post their ideas to get investors. And investors can easily scroll the list and choose the idea they want to invest in. With this application investors and idea generators both can greatly save their time and effort.

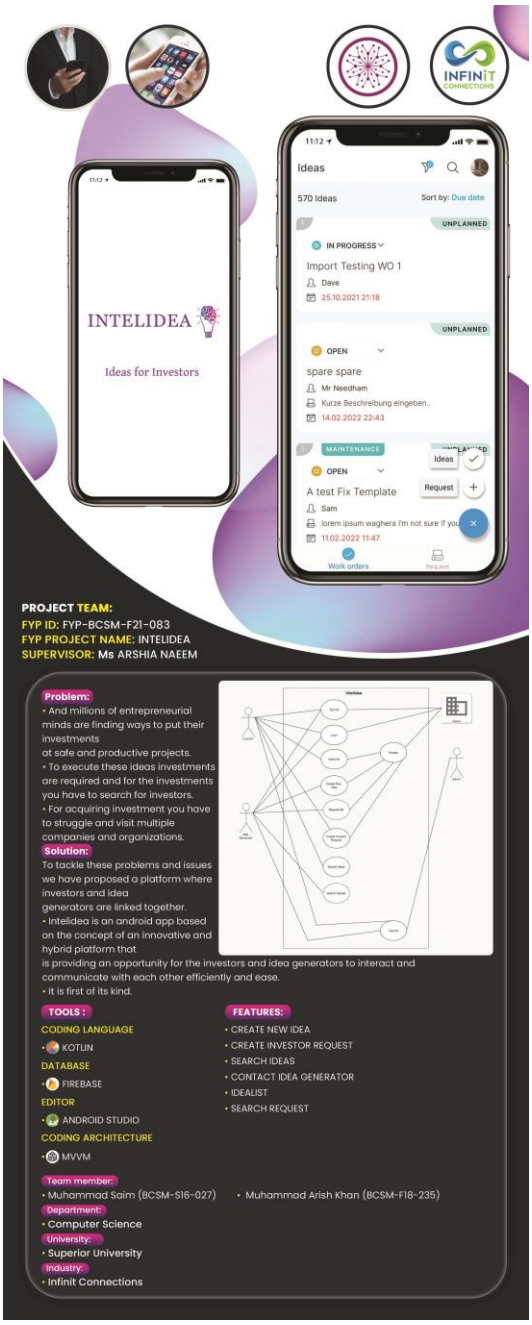
Appendix B

An appendix offers supplemental information that is not necessary for understanding the text itself but may be useful in providing a more thorough grasp of the study subject. It may also contain information that would be too time-consuming to include in the paper's main body.

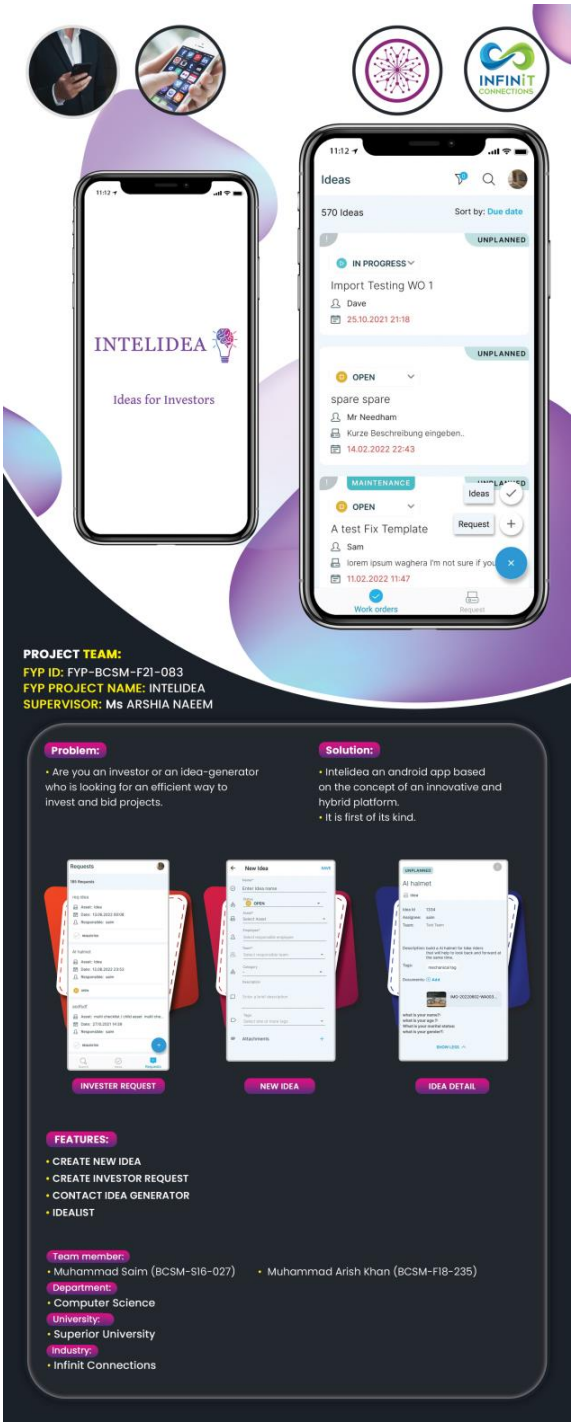
C.1. Broacher



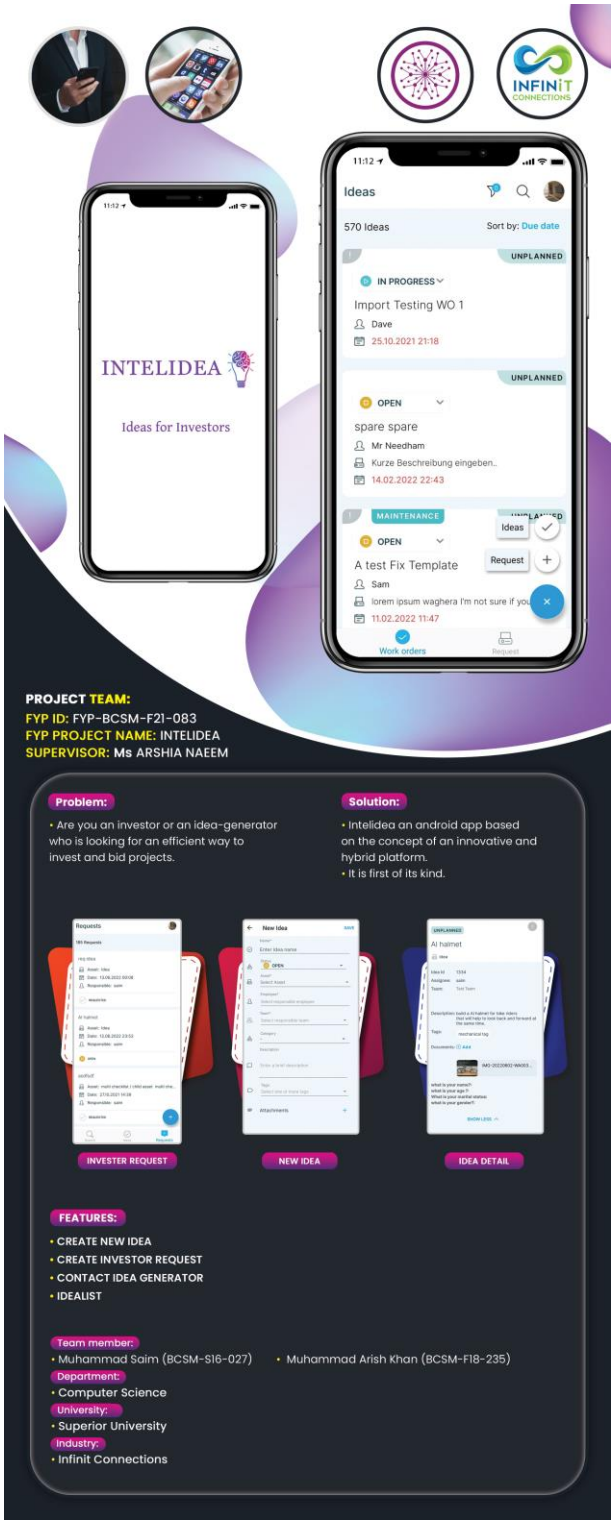
C.2. Flyer


INTELIDEA
Ideas for Investors
Ideas
570 Ideas
Sort by: Due date
UNPLANNED
IN PROGRESS
Import Testing WO 1
Dave
25.10.2021 21:18
UNPLANNED
OPEN
spare spare
Mr Needham
Kurze Beschreibung eingeben..
14.02.2022 22:43
MAINTENANCE
OPEN
A test Fix Template
Sam
11.02.2022 11:47
Ideas
Request
PROJECT TEAM:
FYP ID: FYP-BCSM-F2I-083
FYP PROJECT NAME: INTELIDEA
SUPERVISOR: Ms ARSHIA NAEEM
Problem:
• And millions of entrepreneurial minds are finding ways to put their investments at safe and productive projects.
• To execute these ideas investments are required and for the investments you have to search for investors.
• For acquiring investment you have to struggle and visit multiple companies and organizations.
Solution:
To tackle these problems and issues we have proposed a platform where investors and idea generators are linked together.
• Intelidea is an android app based on the concept of an innovative and hybrid platform that is providing an opportunity for the investors and idea generators to interact and communicate with each other efficiently and ease.
• It is first of its kind.
TOOLS :
CODING LANGUAGE
• KOTLIN
DATABASE
• FIREBASE
EDITOR
• ANDROID STUDIO
CODING ARCHITECTURE
• MVVM
FEATURES:
• CREATE NEW IDEA
• CREATE INVESTOR REQUEST
• SEARCH IDEAS
• CONTACT IDEA GENERATOR
• IDEALIST
• SEARCH REQUEST
Team member:
• Muhammad Saim (BCSM-S16-027)
• Muhammad Arish Khan (BCSM-F18-235)
Department:
• Computer Science
University:
• Superior University
Industry:
• Infinit Connections

C.3. Standee


INTELIDEA
Ideas for Investors
Ideas
570 Ideas
Sort by: Due date
UNPLANNED
IN PROGRESS
Import Testing WO 1
Dave
25.10.2021 21:18
UNPLANNED
OPEN
spare spare
Mr Needham
Kurze Beschreibung eingeben..
14.02.2022 22:43
MAINTENANCE
OPEN
A test Fix Template
Sam
11.02.2022 11:47
Work orders
Request
PROJECT TEAM:
FYP ID: FYP-BCSM-F21-083
FYP PROJECT NAME: INTELIDEA
SUPERVISOR: Ms ARSHIA NAEEM
Problem:
• Are you an investor or an idea-generator who is looking for an efficient way to invest and bid projects.
Solution:
• Intelidea an android app based on the concept of an innovative and hybrid platform.
• It is first of its kind.
INVESTER REQUEST
NEW IDEA
IDEA DETAIL
FEATURES:
• CREATE NEW IDEA
• CREATE INVESTOR REQUEST
• CONTACT IDEA GENERATOR
• IDEALIST
Team member:
• Muhammad Saim (BCSM-S16-027)
• Muhammad Arish Khan (BCSM-F18-235)
Department:
• Computer Science
University:
• Superior University
Industry:
• Infinit Connections

C.4. Banner


INTELIDEA
Ideas for Investors
Ideas
570 Ideas
Sort by: Due date
UNPLANNED
IN PROGRESS
Import Testing WO 1
Dave
25.10.2021 21:18
UNPLANNED
OPEN
spare spare
Mr Needham
Kurze Beschreibung eingeben..
14.02.2022 22:43
MAINTENANCE
OPEN
A test Fix Template
Sam
11.02.2022 11:47
Work orders
Request
PROJECT TEAM:
FYP ID: FYP-BCSM-F21-083
FYP PROJECT NAME: INTELIDEA
SUPERVISOR: Ms ARSHIA NAEEM
Problem:
• Are you an investor or an idea-generator who is looking for an efficient way to invest and bid projects.
Solution:
• Intelidea an android app based on the concept of an innovative and hybrid platform.
• It is first of its kind.
INVESTER REQUEST
NEW IDEA
IDEA DETAIL
FEATURES:
• CREATE NEW IDEA
• CREATE INVESTOR REQUEST
• CONTACT IDEA GENERATOR
• IDEALIST
Team member:
• Muhammad Saim (BCSM-S16-027)
• Muhammad Arish Khan (BCSM-F18-235)
Department:
• Computer Science
University:
• Superior University
Industry:
• Infinit Connections

Reference and Bibliography

Reference and Bibliography

[1] Jack Foster, "*How To Get Ideas*" Research On Idea Generation, 2018 Edition 2, 1/1/18, Berreit Koehler Publishers, Inc/San Francisco, USA, 2018.

[2] Brad M.Barber, "*The Behavior of Individual*" Handbook of the Economics of Finance, 2019 Volume2 Part B, 8/2/19, Alison Publishers, Inc/New Jersey, USA, 2019.

[3] Meir Statman, "*What Do Investors Want?*" The journal of portfolio management, 2020 Edition 1, 12/8/20, Pageant Media Ltd, Inc/Ohio, USA, 2020.

[4] Meir Statman, "*What Do Investors Want?*" The journal of portfolio management, 2020 Edition 1, 12/8/20, Pageant Media Ltd, Inc/Ohio, USA, 2020.

[5] Jack Foster, "*How To Get Ideas*" Research On Idea Generation, 2018 Edition 2, 1/1/18, Berreit Koehler Publishers, Inc/San Francisco, USA, 2018.

[6] Brad M.Barber, "*The Behavior of Individual*" Handbook of the Economics of Finance, 2019 Volume2 Part B, 8/2/19, Alison Publishers, Inc/New Jersey, USA, 2019.

[7] Meir Statman, "*What Do Investors Want?*" The journal of portfolio management, 2020 Edition 1, 12/8/20, Pageant Media Ltd, Inc/Ohio, USA, 2020.

[8] Meir Statman, "*What Do Investors Want?*" The journal of portfolio management, 2020 Edition 1, 12/8/20, Pageant Media Ltd, Inc/Ohio, USA, 2020.

Index