Crowd Funding App

Tony Mathen   W162979.

Jagadeesh Marali   W1641408

Abhishek Muttah   W1649847

Requirement Analysis

User

- User should be able to register to the portal and login using email. Authentication should be implemented for all the users to use the application.
- User should be able to create a project or fund other projects as well as share the project with others.
- User should be able to view all the active projects, the projects he or she has supported, and all the transactions of the amounts funded.

Project

- Every project must have a title, description, funding goal, image and at least one reward.
- There should be option to share the project using the URL to social media sites.
- Once the project is funded, the user should request the admin to disburse funds collected for a project.

Reward

- Every project can have one or more rewards, and the user should not be allowed to fund his or her own project on the platform to reach the funding goal.
- Once the funding goal is reached, the fundraiser can view all the supporters and set the reward status for logging purposes

Usecase Diagram
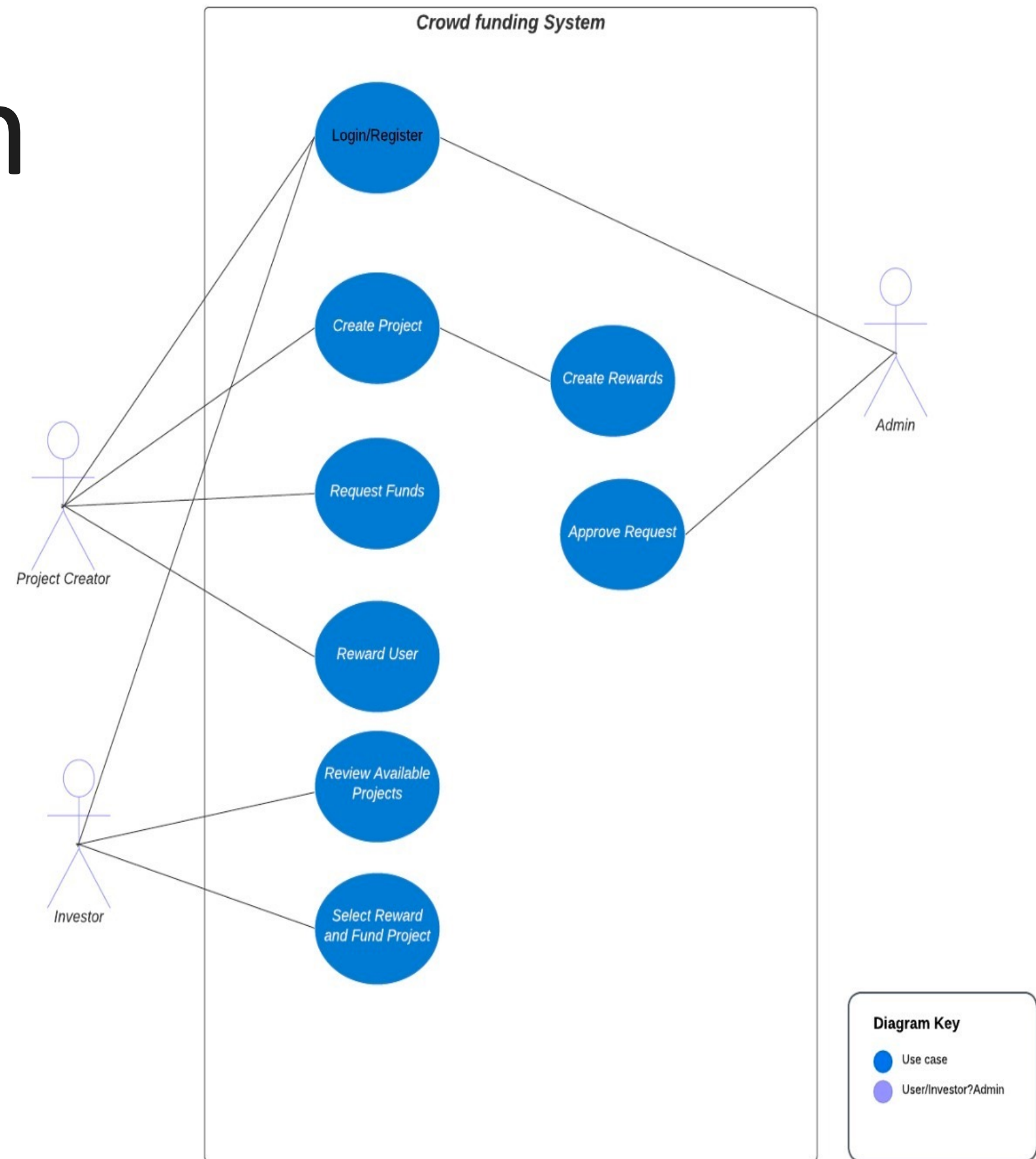
Assumptions

- A user can start his own project as well fund other projects

- A user cannot fund his own project on the platform

- The fundraiser creates a predefined set of amounts that can be donated

- A user has to raise a request to the admin to collect funds of his project

- Once the request is approved by the admin, it cannot be recovered

- The campaign amount would be disbursed to the fundraiser by approval of admin

- If the funding goal is not reached by the campaign end date, the amount collected would be refunded

Classes

BaseModel

The base class for all the classes. Data members are uid, createdAt and updatedAt

Project

Represents the crowd funding project has details like the launch date, funding goal, etc.

Reward

Used with Project class. Every Project has one or more rewards associated with them

Transaction

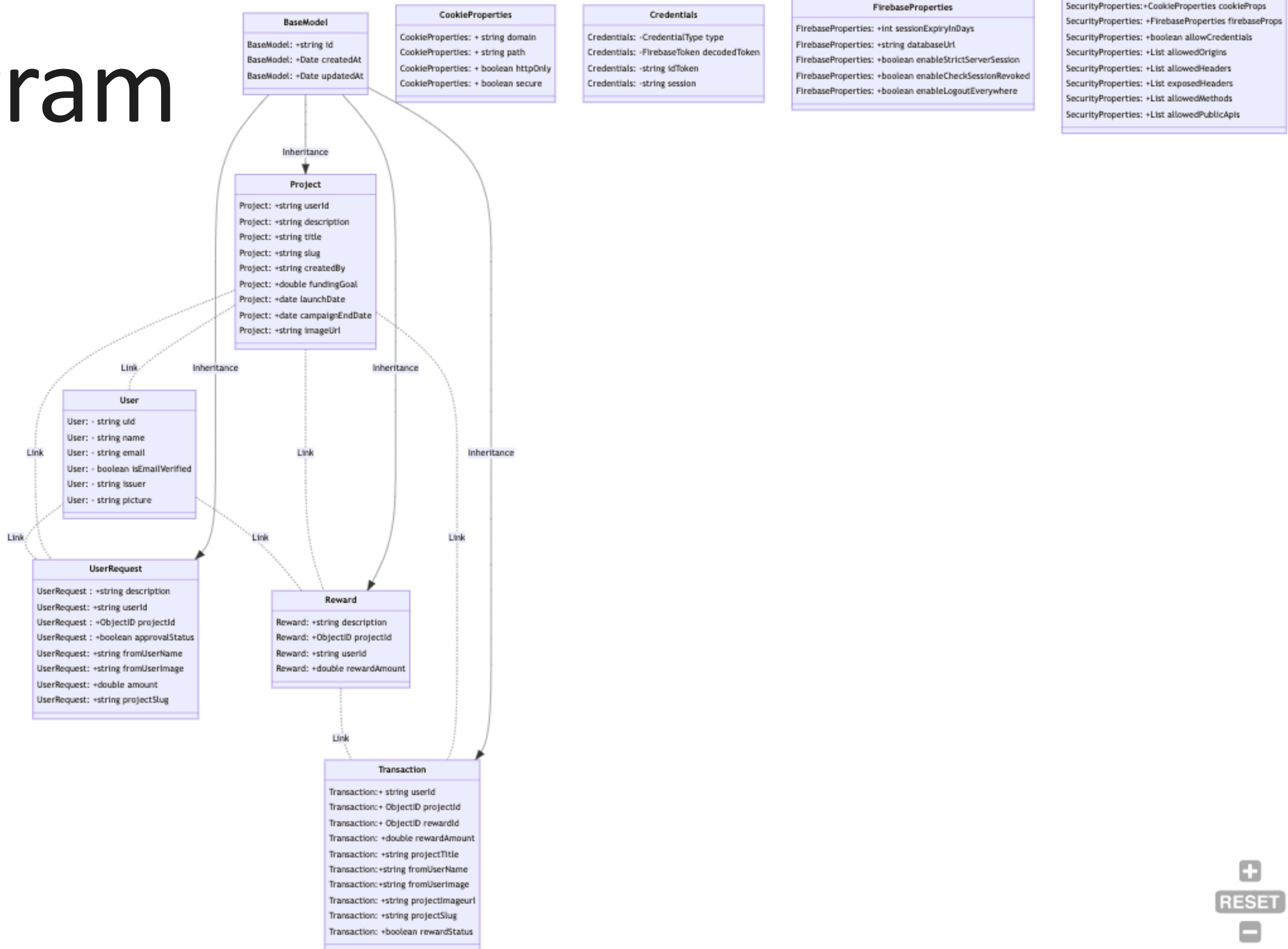Maintains all the fund transactions of the users

User

Maintains user details and is used for authentication

UserRequest

Used with the User class to raise request to disburse funds after the campaign

Class Diagram

BaseModel

BaseModel: +string id
BaseModel: +Date createdAt
BaseModel: +Date updatedAt

CookieProperties

CookieProperties: + string domain
CookieProperties: + string path
CookieProperties: + boolean httpOnly
CookieProperties: + boolean secure

Credentials

Credentials: -CredentialType type
Credentials: -FirebaseToken decodedToken
Credentials: -string idToken
Credentials: -string session

FirebaseProperties

FirebaseProperties: +int sessionExpiryInDays
FirebaseProperties: +string databaseUrl
FirebaseProperties: +boolean enableStrictServerSession
FirebaseProperties: +boolean enableCheckSessionRevoked
FirebaseProperties: +boolean enableLogoutEverywhere

SecurityProperties

SecurityProperties:+CookieProperties cookieProps
SecurityProperties: +FirebaseProperties firebaseProps
SecurityProperties: +boolean allowCredentials
SecurityProperties: +List allowedOrigins
SecurityProperties: +List allowedHeaders
SecurityProperties: +List exposedHeaders
SecurityProperties: +List allowedMethods
SecurityProperties: +List allowedPublicApis

Project

Project: +string userId
Project: +string description
Project: +string title
Project: +string slug
Project: +string createdBy
Project: +double fundingGoal
Project: +date launchDate
Project: +date campaignEndDate
Project: +string imageUrl

User

User: - string uid
User: - string name
User: - string email
User: - boolean isEmailVerified
User: - string issuer
User: - string picture

UserRequest

UserRequest : +string description
UserRequest: +string userId
UserRequest : +ObjectID projectId
UserRequest : +boolean approvalStatus
UserRequest: +string fromUserName
UserRequest: +string fromUserImage
UserRequest: +double amount
UserRequest: +string projectSlug

Reward

Reward: +string description
Reward: +ObjectID projectId
Reward: +string userId
Reward: +double rewardAmount

Transaction

Transaction:+ string userId
Transaction:+ ObjectID projectId
Transaction:+ ObjectID rewardId
Transaction: +double rewardAmount
Transaction: +string projectTitle
Transaction:+string fromUserName
Transaction:+string fromUserImage
Transaction: +string projectImageurl
Transaction: +string projectSlug
Transaction: +boolean rewardStatus

Design Decisions

Repository Pattern

- Provides flexible architecture

- Help us implement the features of DRY(Do not Repeat Yourself)

Denormalization

- Reduce calls to the DB and to improve

OOPS Concepts

- Encapsulation   *getters and setters, con*

- Abstraction   *extending **MongoRepository** interface*

- Inheritance   *BaseModel Class*

- Composition   *Util Library*

Authentication

Firebase

JWT tokens

Authentication

Security Filter

By extending the
OnePerRequestFilter *class*

```java
private void verifyToken(HttpServletRequest request) {
    String session = null;
    FirebaseToken decodedToken = null;
    Credentials.CredentialType type = null;
    boolean strictServerSessionEnabled = securityProps.getFirebaseProps().isEnableStrictServerSession();
    Cookie sessionCookie = cookieUtils.getCookie( name: "session");
    String token = securityService.getBearerToken(request);
    logger.info(token);
    try {
        if (sessionCookie != null) {
            session = sessionCookie.getValue();
            decodedToken = FirebaseAuth.getInstance().verifySessionCookie(session,
                    securityProps.getFirebaseProps().isEnableCheckSessionRevoked());
            type = Credentials.CredentialType.SESSION;
        } else if (!strictServerSessionEnabled) {
            if (token != null && !token.equalsIgnoreCase( anotherString: "undefined")) {
                decodedToken = FirebaseAuth.getInstance().verifyIdToken(token);
                type = Credentials.CredentialType.ID_TOKEN;
            }
        }
    } catch (FirebaseAuthException e) {
        e.printStackTrace();
        log.error("Firebase Exception:: ", e.getLocalizedMessage());
    }
    User user = firebaseTokenToUserDto(decodedToken);
    if (user != null) {
        UsernamePasswordAuthenticationToken authentication = new UsernamePasswordAuthenticationToken(user,
                new Credentials(type, decodedToken, token, session),  authorities: null);
        authentication.setDetails(new WebAuthenticationDetailsSource().buildDetails(request));
        SecurityContextHolder.getContext().setAuthentication(authentication);
    }
}
```

Frontend

Architecture

dependencies
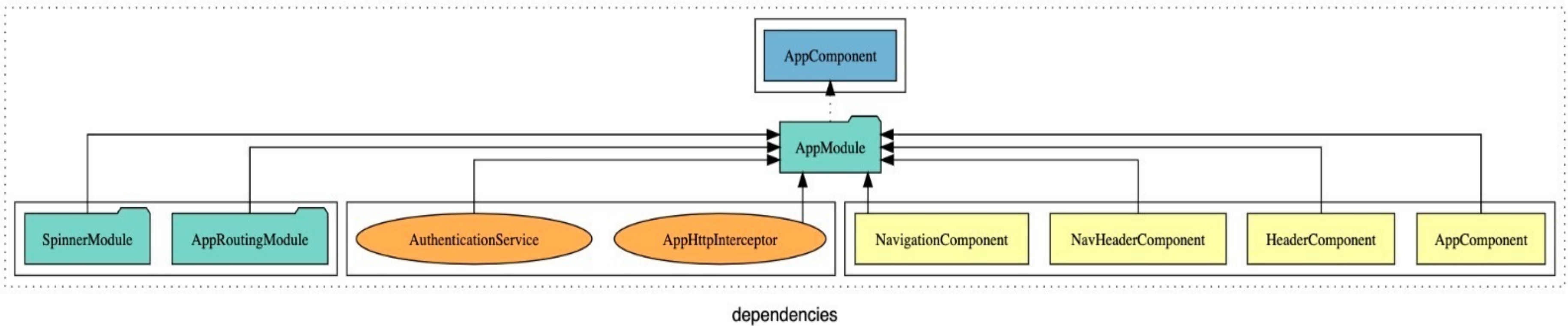
Sequence Diagram
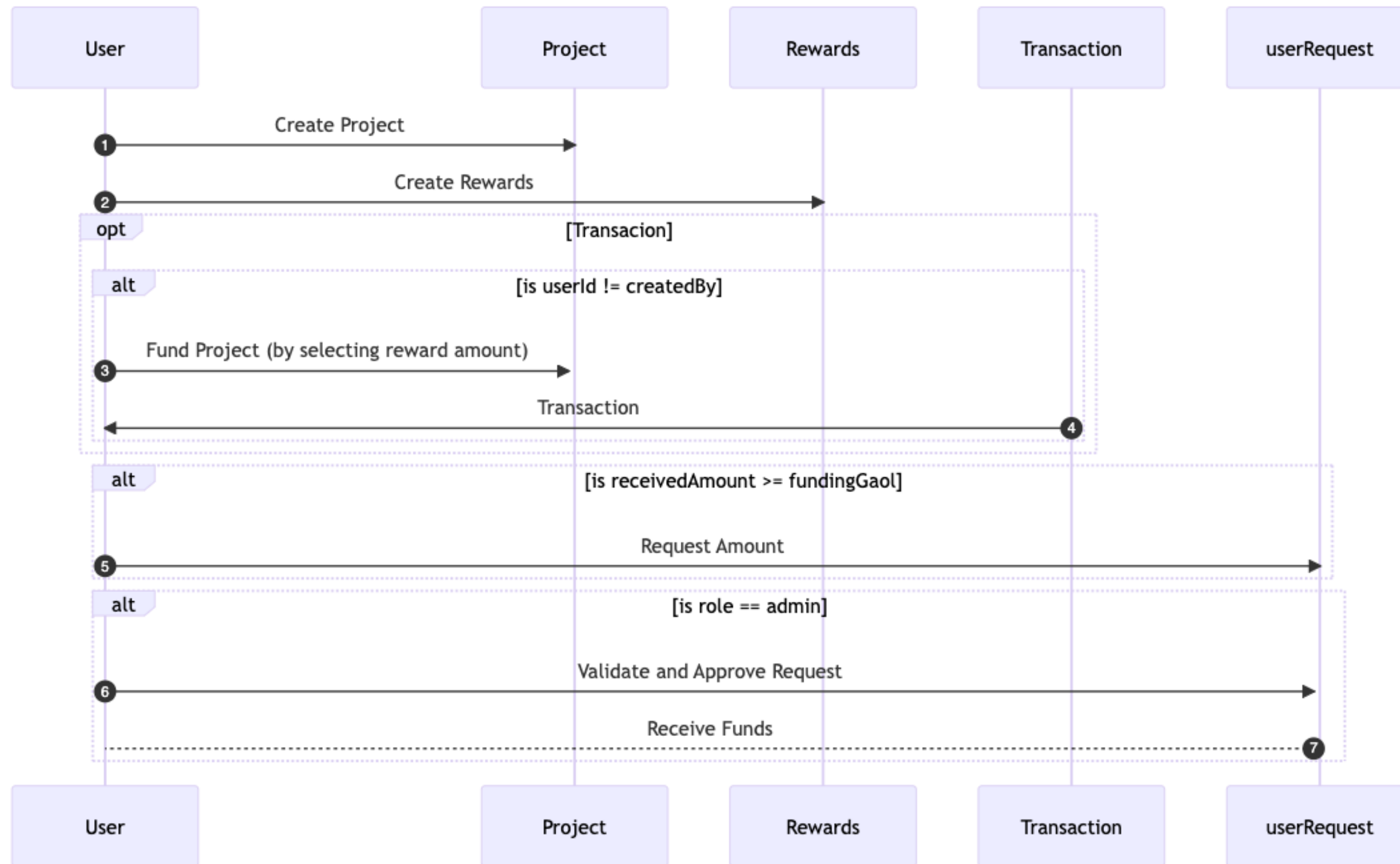
Screenshots

Sign up your account

Email

hello@example.com

Password

Sign me up

Already have an account? **Sign in**

Sign in your account

Email

Password

Forgot Password?

Sign Me In

Don't have an account? **Sign up**

G **Login with Google**

Screenshots

Screenshots

Luji's Chocolate : Made Directly with Farmers in West Africa

$ 13240

Availability: 30/12/2022

Created By: Tony Mathen

Amount Received: $ 26

The Next Generation of Chocolate Our goal: create the best tasting chocolate for you while generating economic opportunities in West African farming communities. We believe chocolate is the most soothing and delicious treat out there, plain and simple. That's why we're on a mission to make it better for you & all players who help get it to you!

Description:
Luji's Backer-Only Tee Choose between the "Chocolate is My Love Language" Tee & "Doing Good Should Taste Good" Tee

Amount: 26

Description:
Luji's Sampler - Early Bird 4 chocolate bars in our gift library including a tasting guide for you & we'll deliver 5 seedlings to a cocoa farmer.

Amount: 37

Copy link   Facebook   LinkedIn   Twitter   WhatsApp

Screenshots

Screenshots

Crowd Funding App

t **tony mathen**
tonymathen@gmail.com

Transaction ID	Date	Project Title	Amount	Note	Reward Status
63816d5cc207583a7fd7355f	25/11/2022	Luji's Chocolate : Made Directly with Farmers in West Africa	26	N/A	**false**

Screenshots

Transaction ID	Date	User	Amount	Note	Reward Status	Action
637c15342f586558ef14117e	21/11/2022	Jagadeesh Marali	10	N/A	**true**	Reward
637c1586fc9e933cffe0c154	21/11/2022	Jagadeesh Marali	50	N/A	**false**	Reward

Screenshots - Admin

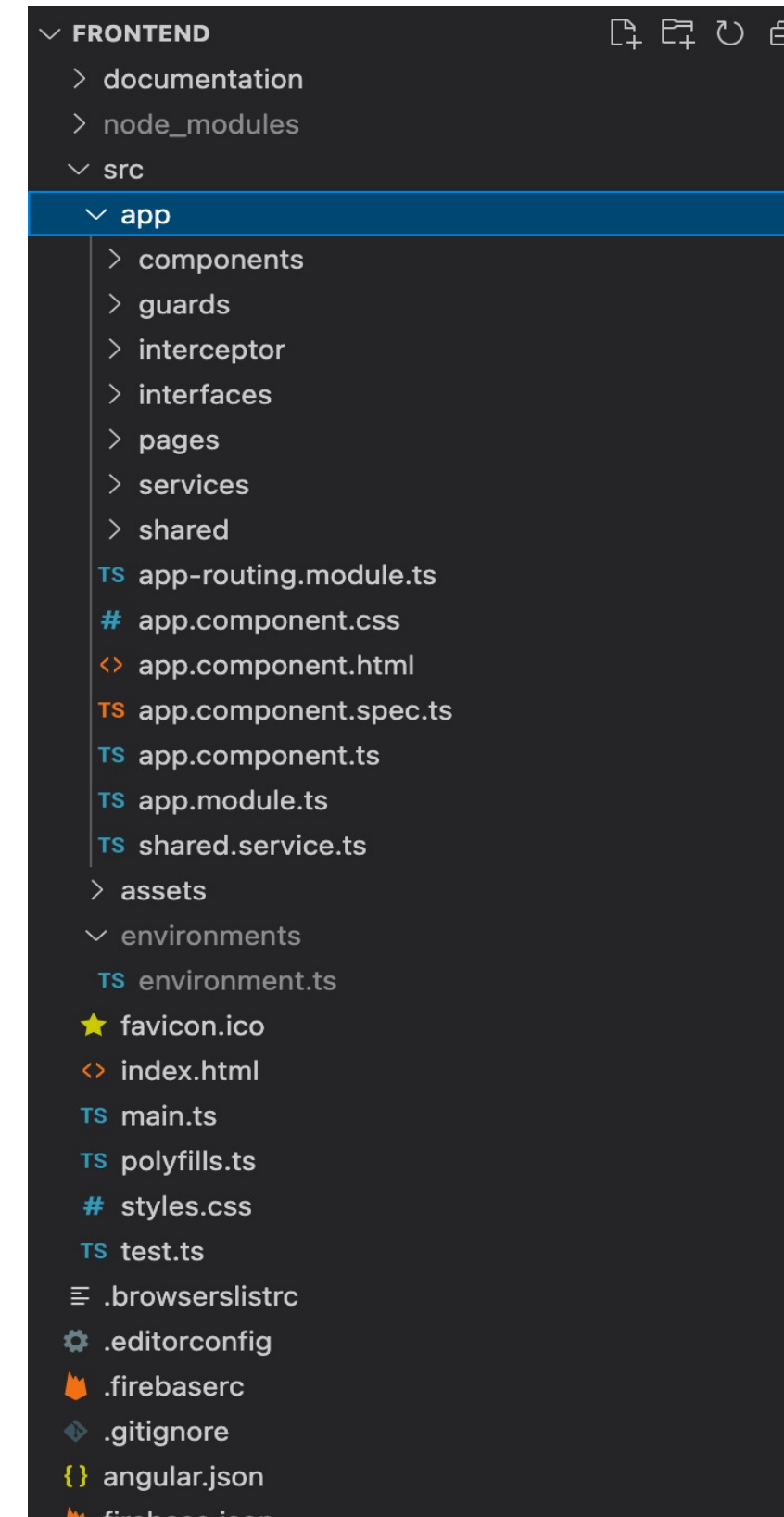| Request ID	Date	Request Description	Amount	From	Request Status	Action
637eae4774a7650c48256422	23/11/2022	Heroes of Might & Magic III The Board Game	100	Jagadeesh Marali	**true**	Approve

Screenshots
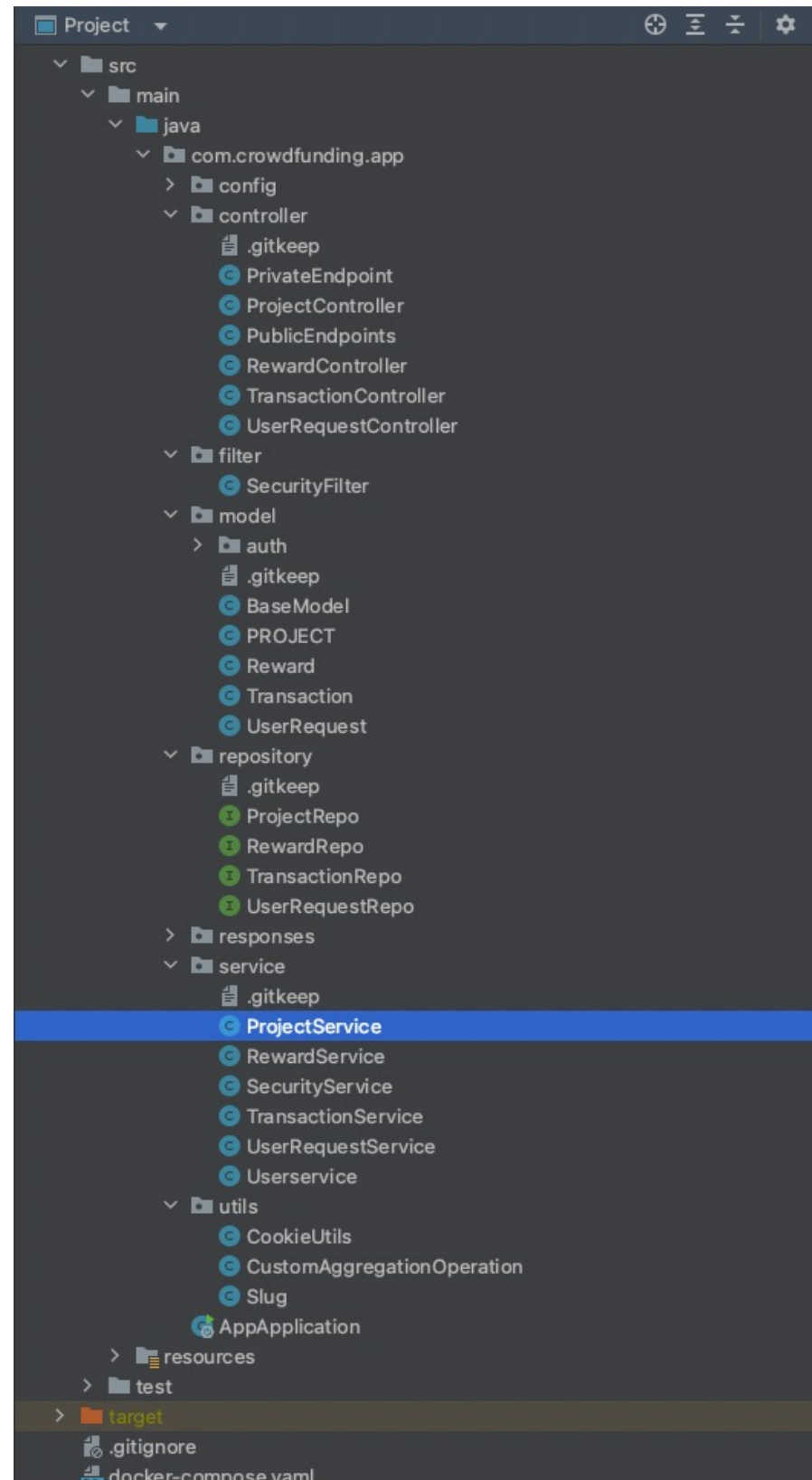
Backend

Frontend

Technology Stack

Database: MongoDB, Firebase

Backend: Java, Springboot

Frontend: HTML/CSS, Typescript, Angular

Tools: IntelliJ, VSCode, Postman, mermaid.js (for UML diagrams)

Deployment:

- **Database:** MongoDB **Atlas**

- **Backend:** Heroku

- **Frontend:** Firebase

URL: https://crowd-funding-app-5b9ee.web.app/

Repo: https://github.com/jagadeeshmarali/crowd-funding-app

Thank you !!