# ERROR REPORTING SYSTEM

***Group Number : 27***

141803008 Prasanna Moon  
141803010 Ronak Patil  
141803011 Saurabh Patil

# ERROR REPORTING

## Abstract

*Software development is both challenging and complex. Specifically, whether we're developing landing page apps or Linux kernel drivers, developing software often requires recording and analysis of vast swathes of information, application exceptions are essential. Well,if you want to find out why a user encountered a problem, why an error occurred, which error is more frequent,and why an application crashed.Every developer writes tons of bugs. And many of those bugs get shipped to production. That's unavoidable.Error reporting allows developers to easily discover and view the errors that their application may be generating. By surfacing error information where the code is being developed, efficiency and awareness can be increased.*

## Introduction

Software can run on all kinds of devices: servers, web browsers,mobile and even your watch. The more an application grows the more bugs will appear. Just think of companies like Facebook and Airbnb which have millions of users using their products on a variety of devices. In an industry where you need to move fast it is unavoidable to introduce new software bugs during development. However, you can greatly improve the process of finding and fixing bugs.

Error Monitoring

- Most users won't report errors.
  If you have a B2B application, it is more likely users will report errors since they might be paying you for using it. For B2C applications (e.g. online shops, newspapers) users are less likely to report errors and will just quit using your application instead of notifying you. Especially if the user is using your application for the first time it is easy for them to quit silently.
- Error reports of users are often insufficient.
  "I cannot buy anything, please fix this!" is not a very helpful error report. As a developer you are interested in the error logs, the environment (e.g. browser) and the steps to reproduce the error.
- Some errors are critical and need to be addressed quickly.
  If users cannot login anymore when using the Mozilla Firefox browser this needs to be acknowledged and resolved quickly.
- Error monitoring allows you to better understand why issues happen.
  A pattern could be that 40 percent of all bug reports happen due to using JavaScript features which do not work in older web browsers.
- Software tests won't be able to uncover all issues.
  While having a good test coverage can help a lot to prevent bugs to be shipped to production you cannot possibly test everything. Unit tests won't catch all issues as testing each part individually is different from testing the whole system. End-to-end tests are rather slow to execute so you often only test the happy path

## Aims & Objectives

Deploying fast and frequently might introduce an errors, exceptions and bugs into our application, our ultimate aim is to identify them before your users do. This System will provide end-to-end insights that enable companies to prevent unsafe code from being promoted into production and solve issues quickly in any environment. This system will provide a platform where developers can see and prioritize an errors, exceptions and bugs with an ease in real time.

## Motivation

We were searching for something that will provide more information to developers in real time so they can keep track of their products in a better way. We found out that 90 percent of users do not report errors in real-time which causes a lot of problems for developer to identify and solve the error. We thought of developing a solution to this problem, as a result of our research, we decided to develop Error reporting service which will provide real-time reporting of errors to developer so they now focus on solving the error and not on identification.

## Proposed System

The Report service provides set of tools and software to deal with the errors and logs being thrown by apps used by people worldwide. We will provide the following under ReportService

- A Framework - Used by developer instead of using traditional console logs the framework provides logging the error directly to our server
- The Server - Connects to the database and post all error to the database post validation.
- The App - The app provides the way to see all the errors logged by the apps of the particular company. Each developer of the company gets an account for the usage of the app
- The Database - The mysql database to keep the record of all the companies, their developers(users) and their apps(web or mobile) and the errors logged by these apps

## Hardware & Software Requirements

- Processors: Intel Atom® processor or Intel® Core™ i3 processor.
- Disk space: 1 GB.
- Operating systems: Windows* 7 or later, and Linux.
- MySQL database
- http server connecter
- java script
- flutter and Android studio
- Html, CSS

## Functional Requirements

- Providing error log dashboard
- Providing option to sort error logs according to priority
- Login and profile management for each developer
- Error reporting framework module to import in source code
- Providing manual and automatic insertion of error logs in system

## Non Functional Requirements

- Real time updating of logs over server within 60 seconds
- Providing Manual key sorting of logs
- Providing secure database by tunneling through HTTP server
- Security of accessing logs through password over database
- Providing export system for downloading log files
- The application should be user friendly and not complicated
- Server capacity to handle more than 300 HTTP request to access database between 5 seconds

## Selected Process Model

Incremental Process Model is being used to develop this project from scratch.

- Incremental Process Model is a type of Prescriptive Analysis Model. Here the model prescribes a set of activities, actions, tasks, quality assurance and change the mechanism for every project.
- Incremental Process Model's steps are as given below:
  - Communication - Project Initiation and requirement Gathering
  - Planning - Estimating, Scheduling, tracking
  - Modelling - Analysis and design of overall project.
  - Construction - Coding/ implementation and testing the software.
  - Deployment - Delivery and Feedback

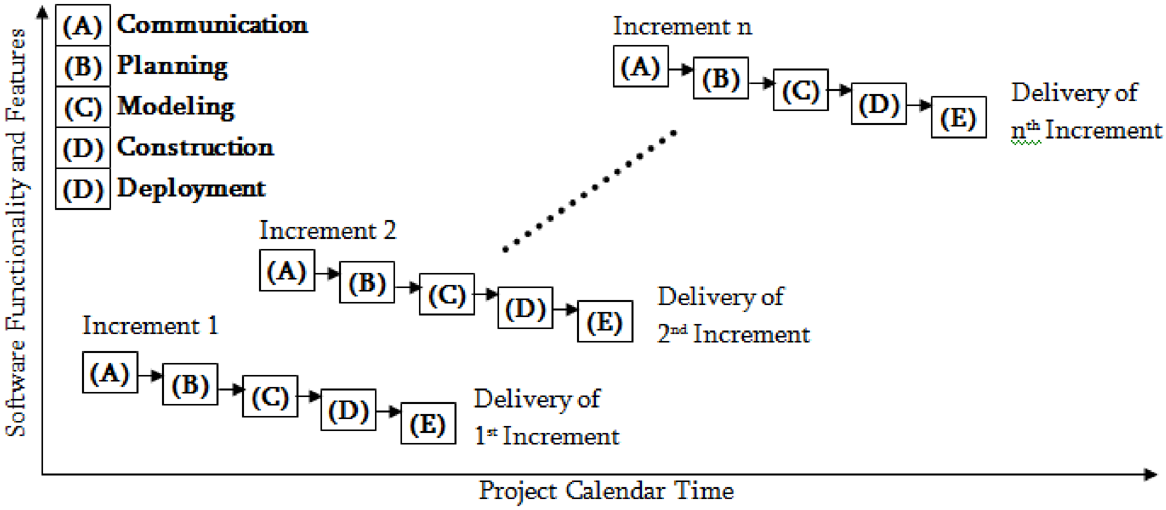
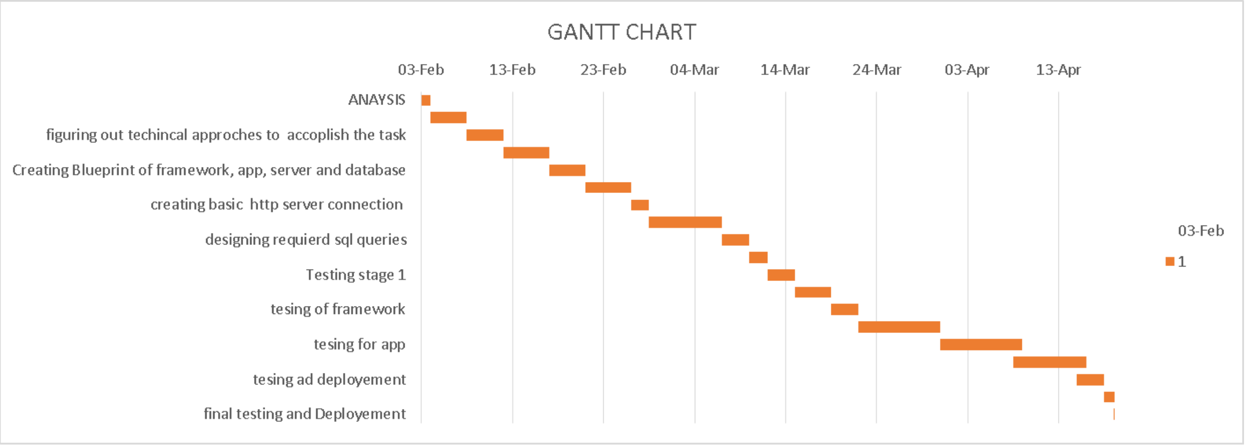


**Figure : Flowchart of Incremental Model**

## Gantt Chart



Proposed System Architecture

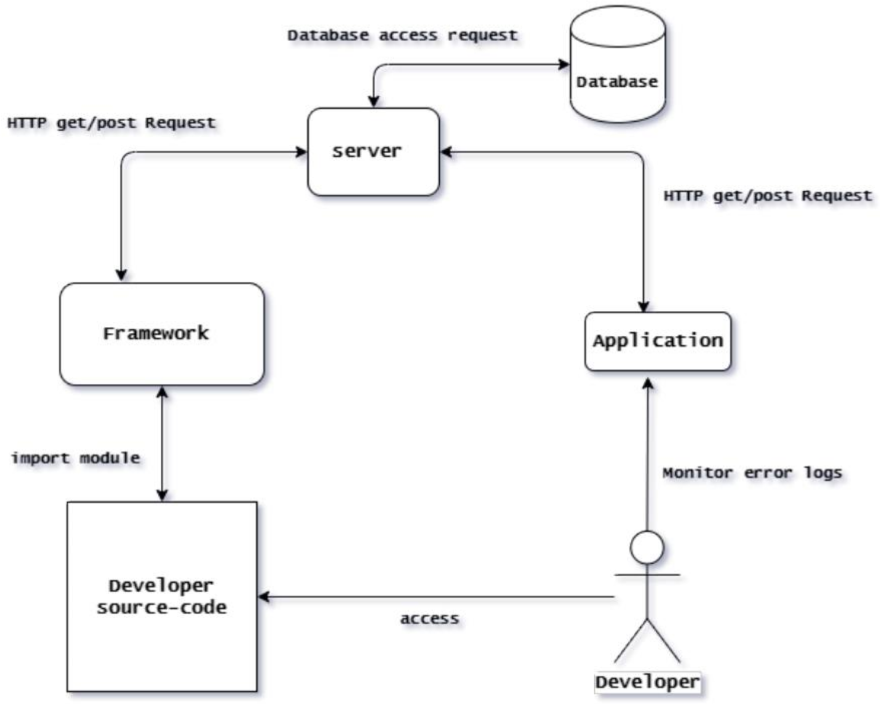
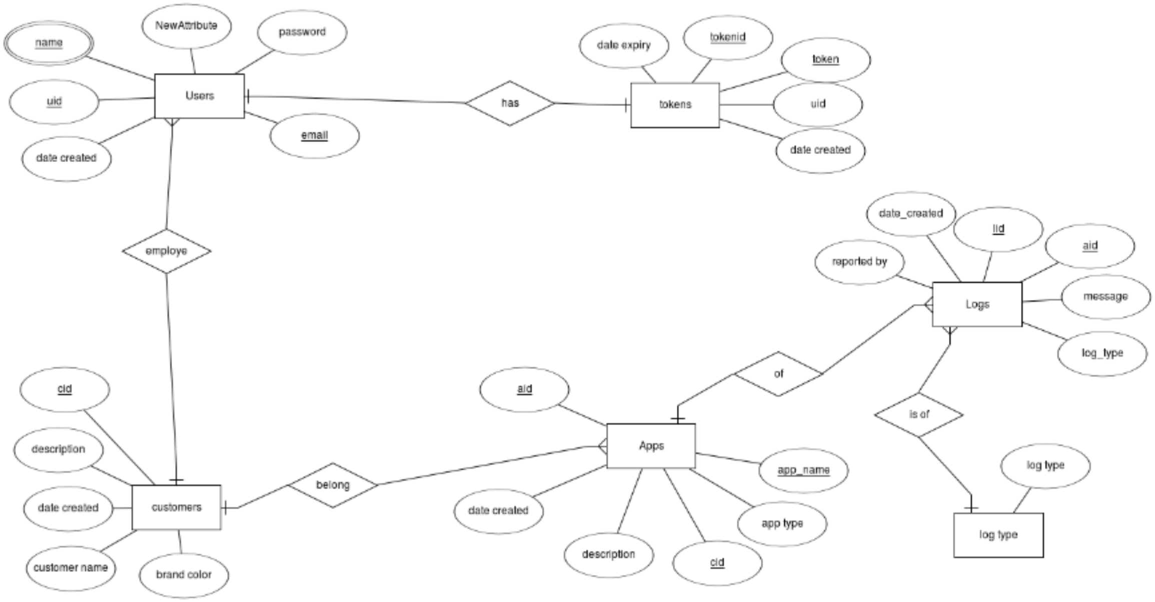
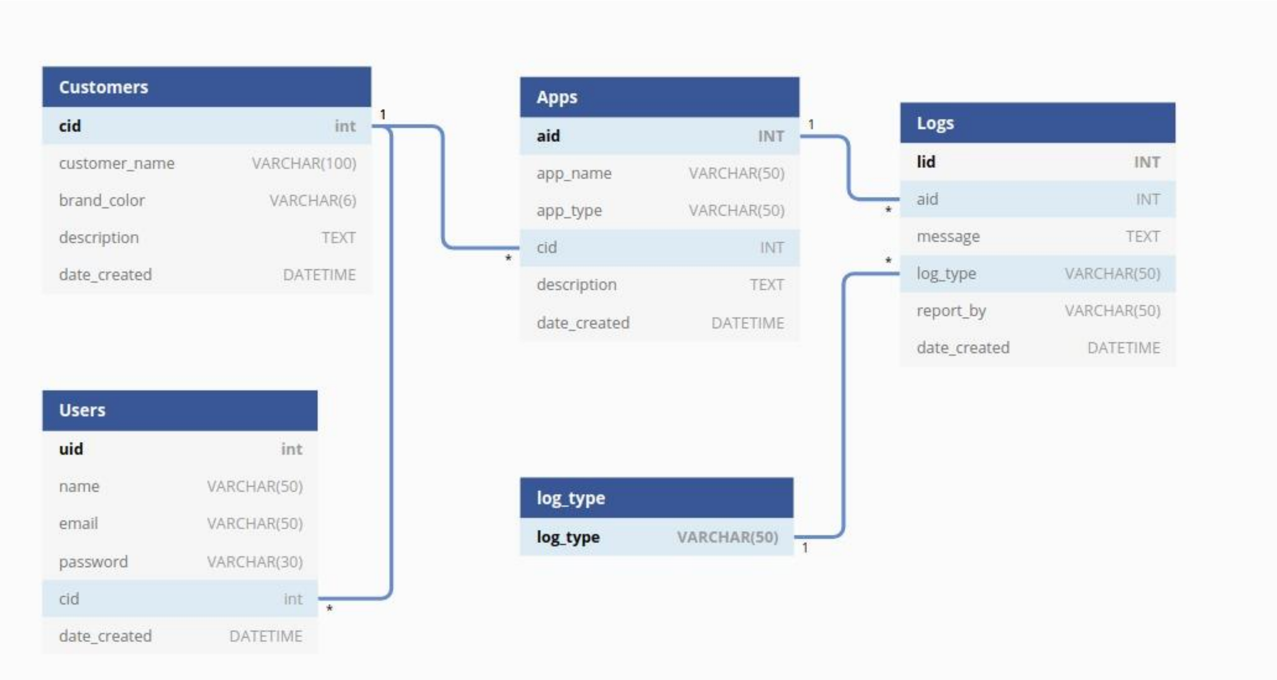


fig 1.1 : Proposed System Architecture

## Entity Relationship Diagram



## Class Schema Diagram



| Customers | |
|---|---|
| **cid** | int |
| customer_name | VARCHAR(100) |
| brand_color | VARCHAR(6) |
| description | TEXT |
| date_created | DATETIME |

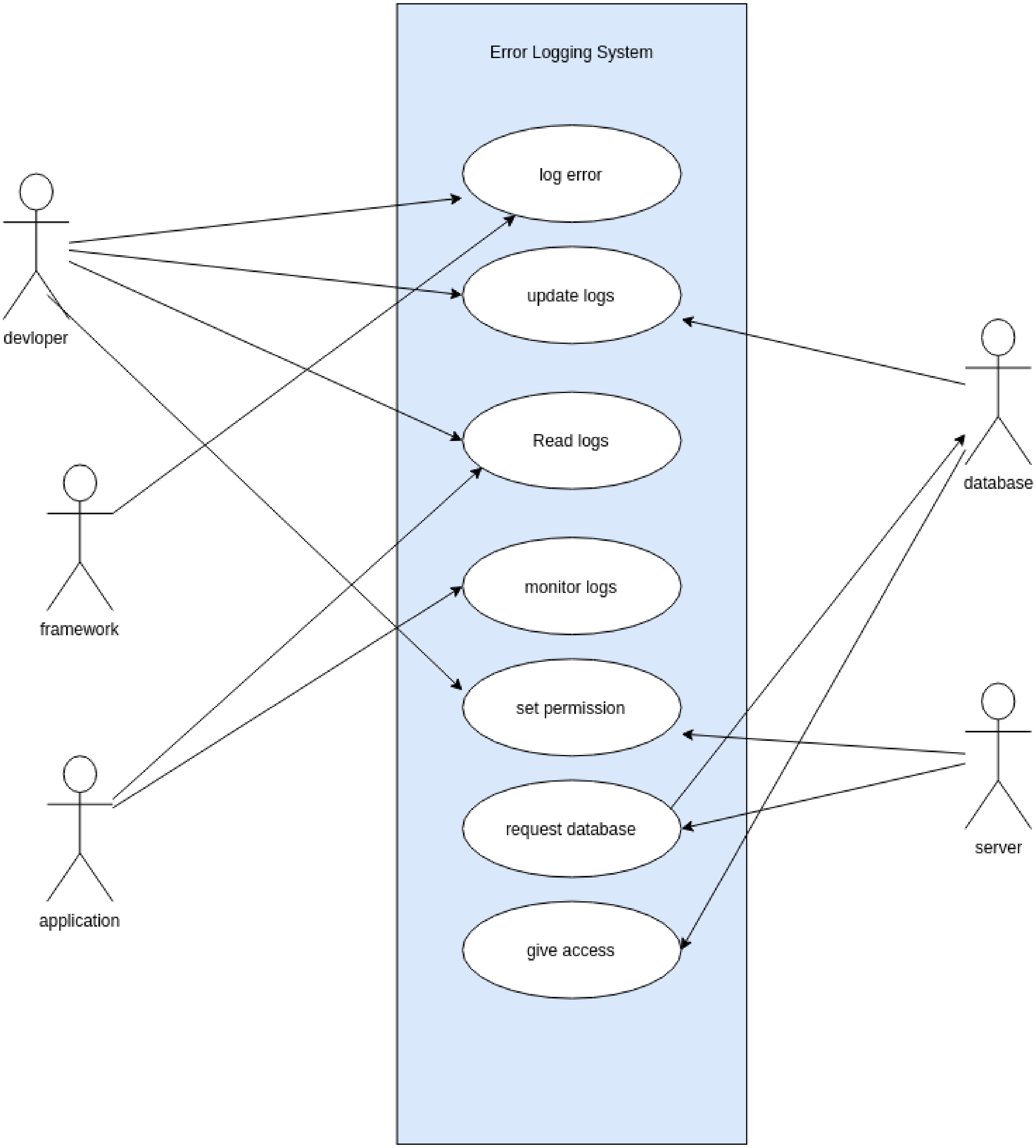
| Apps | |
|---|---|
| **aid** | INT |
| app_name | VARCHAR(50) |
| app_type | VARCHAR(50) |
| cid | INT |
| description | TEXT |
| date_created | DATETIME |

| Logs | |
|---|---|
| **lid** | INT |
| aid | INT |
| message | TEXT |
| log_type | VARCHAR(50) |
| report_by | VARCHAR(50) |
| date_created | DATETIME |

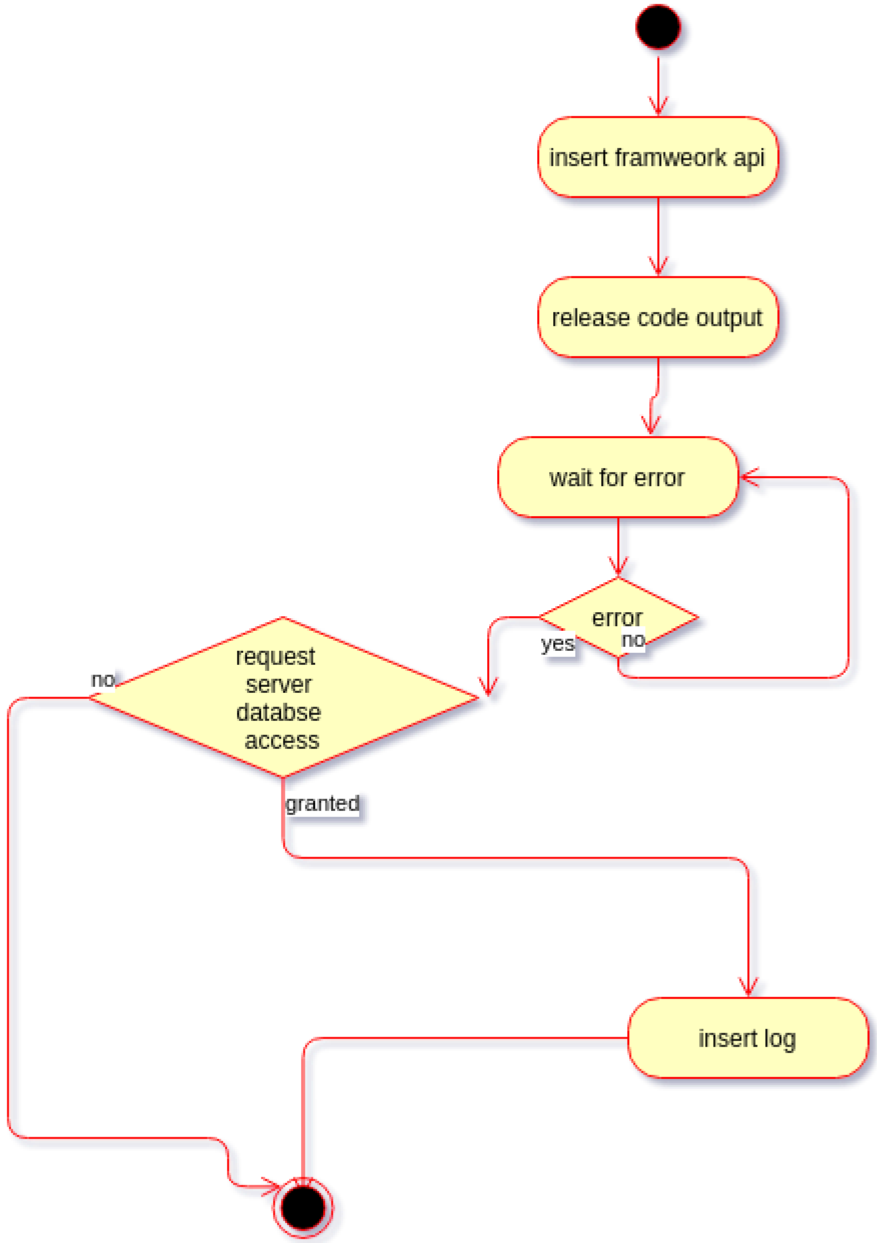
| Users | |
|---|---|
| **uid** | int |
| name | VARCHAR(50) |
| email | VARCHAR(50) |
| password | VARCHAR(30) |
| cid | int |
| date_created | DATETIME |

| log_type | |
|---|---|
| **log_type** | VARCHAR(50) |

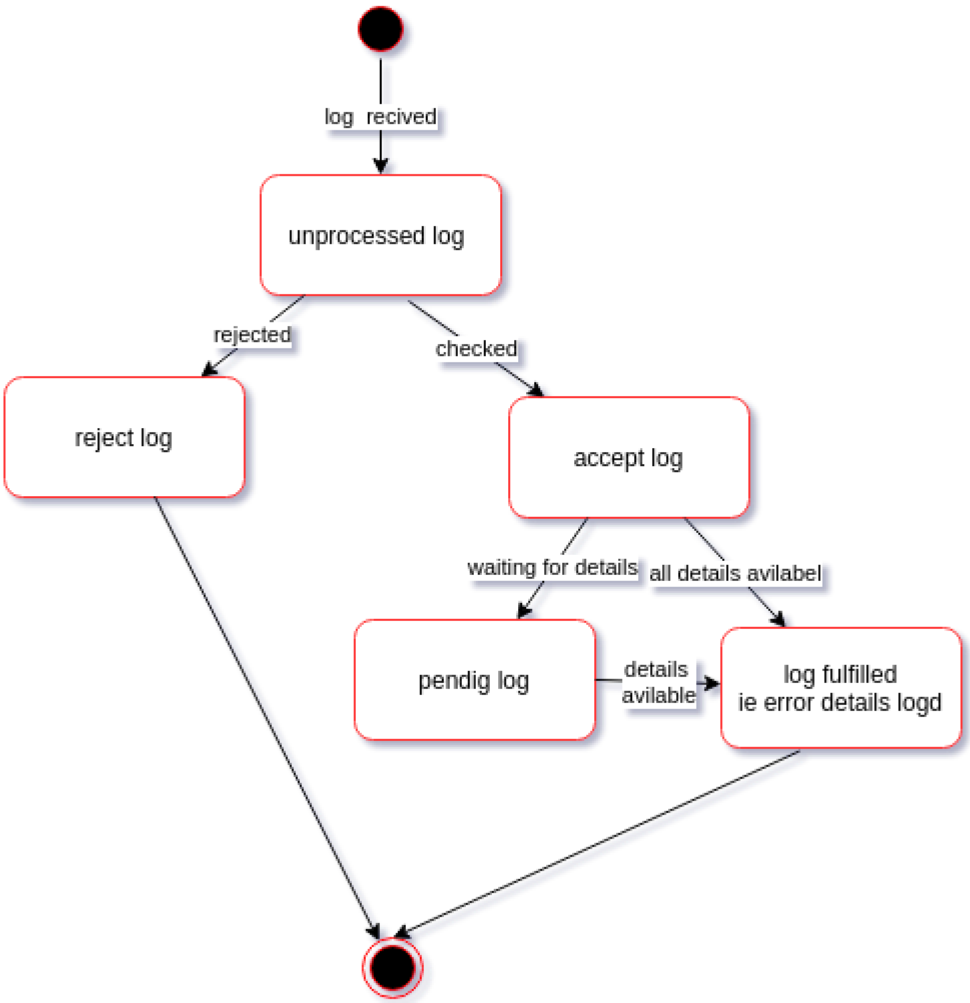
## Use case Diagram



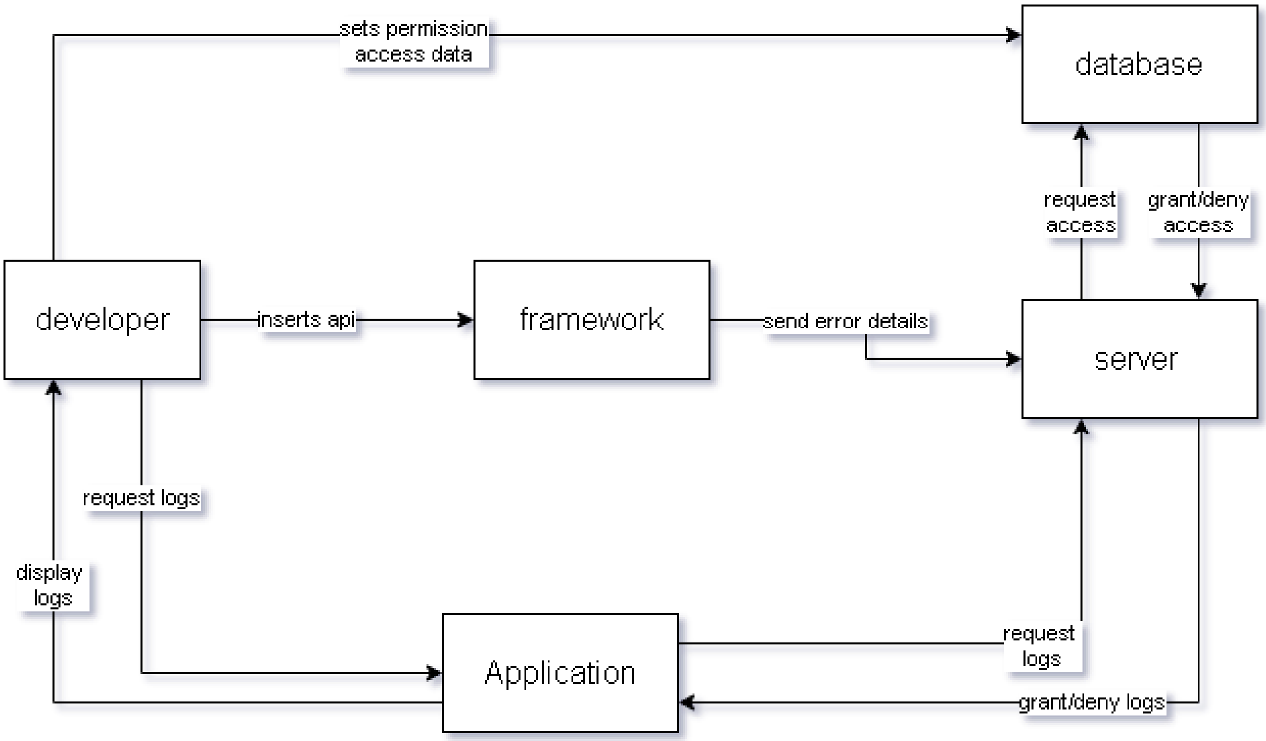
## Activity Diagram



## State Diagram



## Data flow Diagram


sets permission access data
database
request access
grant/deny access
developer
inserts api
framework
send error details
server
request logs
display logs
Application
request logs
grant/deny logs