Low Level Design

Blog Creator Web Application

Written By	Atharva Pimputkar, Harshbhai Solanki and Vishal Rai
Document Version	0.3
Last Revised Date	12 – March -2022

Document Control

Change Record:

Version	Date	Author	Comments
0.1	10 – March - 2022	Vishal Rai	Introduction & Architecture defined
0.2	11 – March - 2022	Atharva Pimputkar	Architecture & Architecture Description appended and updated
0.3	12 – March - 2022	Harshbhai Solanki	Unit Test Cases defined and appended

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

1. Introduction

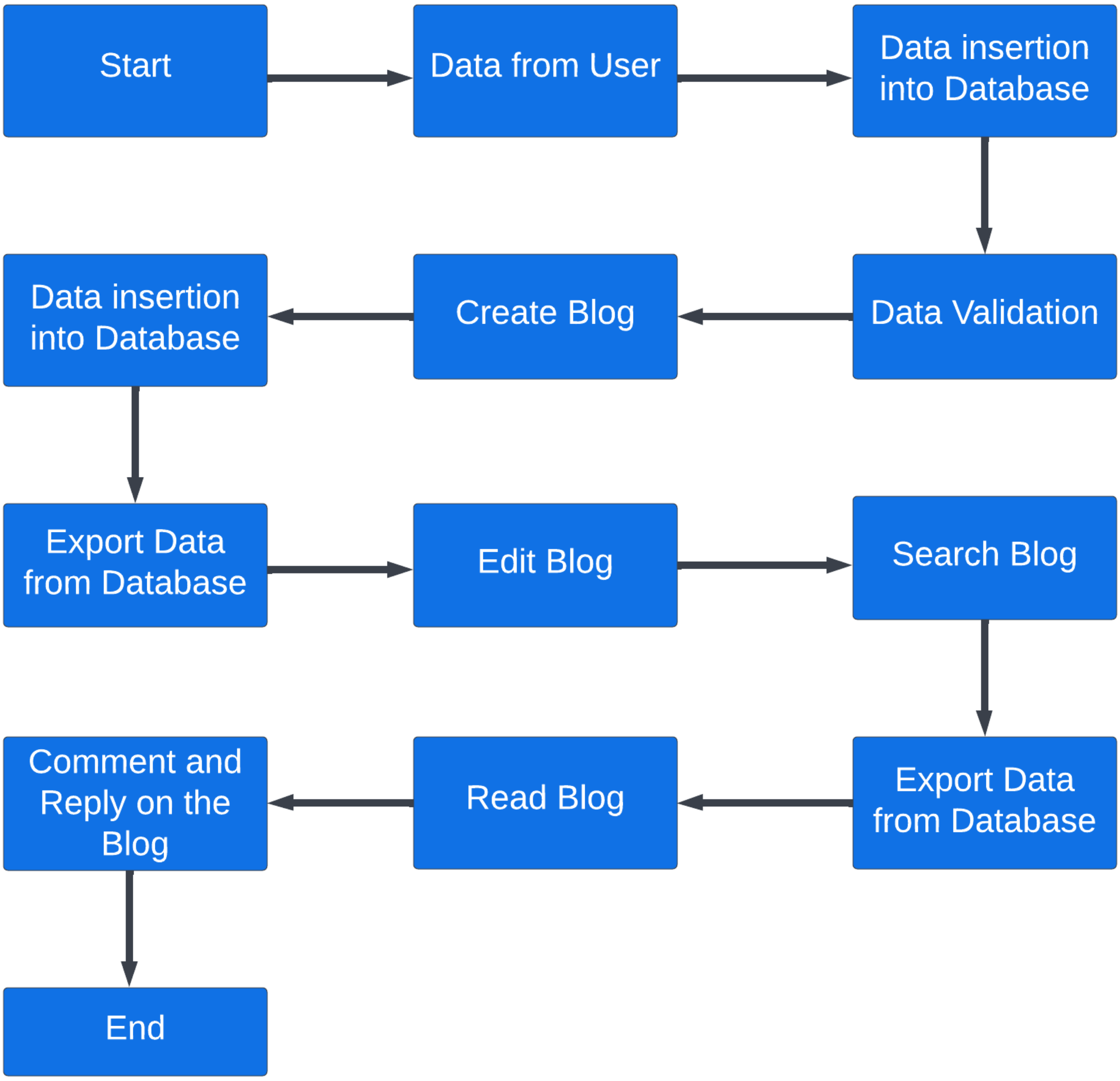
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Blog Creator Web Application. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2. Architecture


Start
Data from User
Data insertion into Database
Data Validation
Create Blog
Data insertion into Database
Export Data from Database
Edit Blog
Search Blog
Export Data from Database
Read Blog
Comment and Reply on the Blog
End

3. Architecture Description

3.1. Data from User

We will collect personal details from users such as first name, last name, username, email address and password during signup.

3.2. Data Insertion into Database

Collecting the data given by the user and storing it into the database. Database is Sqlite.

3.3. Data Validation

Data validation will be done on the data given by the user and if the data matches, the user will be able to login into the blog web application.

3.4. Create Blog

Here users can create a blog by giving the title and content of the blog.

3.5. Data Insertion into Database

Data of the blog given by the user along with the name will be stored in the sqlite database.

3.6. Export Data from Database

All the blogs created by a particular user will be fetched from the sqlite database and exported to the edit page.

3.7. Edit Blog

Users can see all the blogs created by them along with the date of creation of the blog. Users can make any blog private so no other user can be able to view the blog. Users can also delete any blog if not required and can be able to edit any field of their choice in the blog.

3.8. Search Blog

Here we can search any blog by entering a particular keyword.

3.9. Export Data from Database

All the blogs will be fetched from sqlite database any displayed on the blog page.

3.10. Read Blog

Users can view all the blogs and can read any blog from all the listed blogs. They can also see the date and time at which the blog was posted along with the name of the author and number of views on the blog.

3.11. Comment and Reply on the Blog

Users who are logged in can comment on any blog as well as reply to others' comments.

4. Unit Test Cases

Test Case Description	Prerequisite	Expected Result
Verify whether the Application URL is accessible to the user.	1. Application URL should be defined.	Application URL should be accessible to the user.
Verify whether the Application loads completely for the user when the URL is accessed.	1. Application URL is accessible . 2. Application is deployed.	The Application should load completely for the user when the URL is accessed.
Verify whether the User is able to sign up in the application.	1. Application is accessible.	The User should be able to sign up in the application.
Verify whether user is able to successfully login to the application.	1. Application is accessible. 2. User is signed up to the application.	Users should be able to successfully login to the application.
Verify whether user is able to contact.	1. Application is accessible	User should be able to send query.
Verify whether user is able to enter value in search blog input field.	1. Application is accessible.	User should be able to enter the value.
Verify whether user is presented with recommended results on clicking search.	1. Application is accessible.	User should be presented with recommended results on clicking submit.
Verify whether user is able to view all the blogs on the blog page.	1. Application is accessible.	User should be able to view all the blogs.
Verify whether user is able to open blogs and read it.	1. Application is accessible.	User should be able to open a blog and read it.

Verify whether user is able to view the date and time when the blog was posted and number of views currently on the blog.	1. Application is accessible.	User should be able to view the date and time along with the number of views.
Verify whether user is able to see create blog button	1. Application is accessible. 2. User is signed up to the application. 3. User is logged in to the application.	User should be able to see it.
Verify whether user is able to create a blog.	1. Application is accessible. 2. User is signed up to the application. 3. User is logged in to the application.	User should be able to create a blog.
Verify whether user is able to view list of all the blogs created	1. Application is accessible. 2. User is signed up to the application. 3. User is logged in to the application.	User should be able to view the list.
Verify whether user is able to edit or delete the blog.	1. Application is accessible. 2. User is signed up to the application. 3. User is logged in to the application.	User should be able to edit or delete the blog.
Verify whether user is able to view comments and replies on the blog.	1. Application is accessible.	User should be able to view comments and replies
Verify whether user is able to post a comment and reply to others' comments.	1. Application is accessible. 2. User is signed up to the application. 3. User is logged in to the application.	User should be able to post a comment and reply on others' comments.