2nd Phase Project Presentation for Final Year Projects will be scheduled from next Monday 13/04/2020

The format for project evaluation is as follows

i. Design Diagrams – Class Diagram, Activity Diagram, Interaction Diagram, Sequence Diagram etc. (Any 2).

ii. Methodology – Methods, Algorithms used.

iii. Description about Modules with Implementation Details.

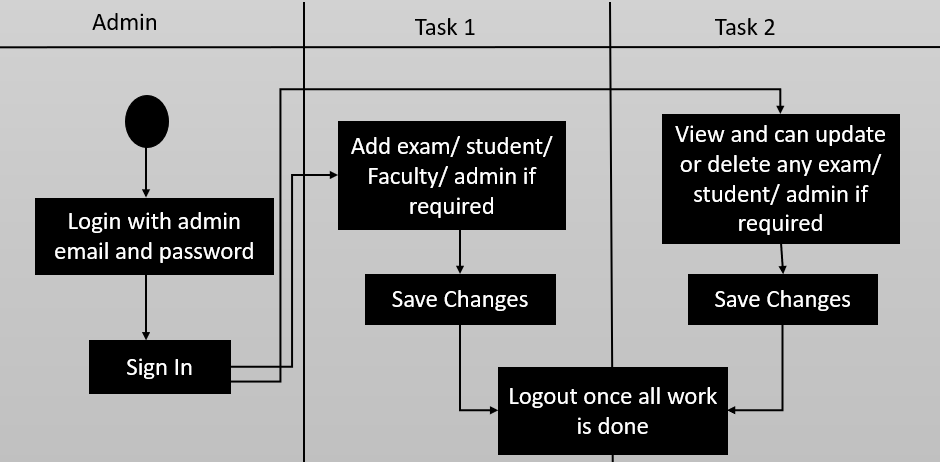
iv. Expected Outcomes.

***Design diagrams***

**Unified Modeling Language (UML) | Activity Diagrams**

We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens.

UML models basically three types of diagrams, namely, structure diagrams, interaction diagrams, and behavior diagrams. An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram.



**Class diagram**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

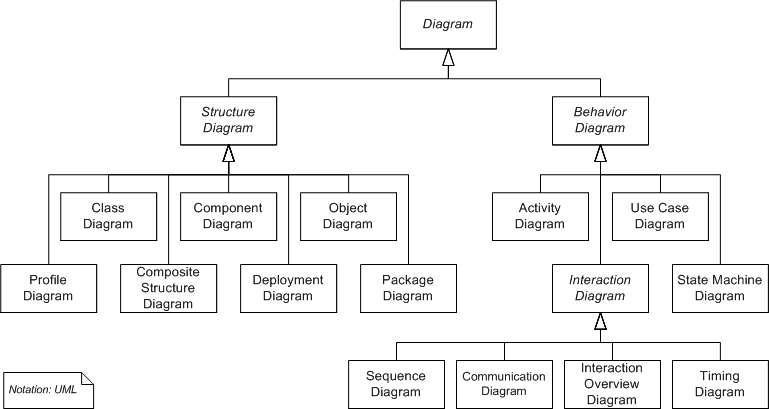
The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

1. The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

2.The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

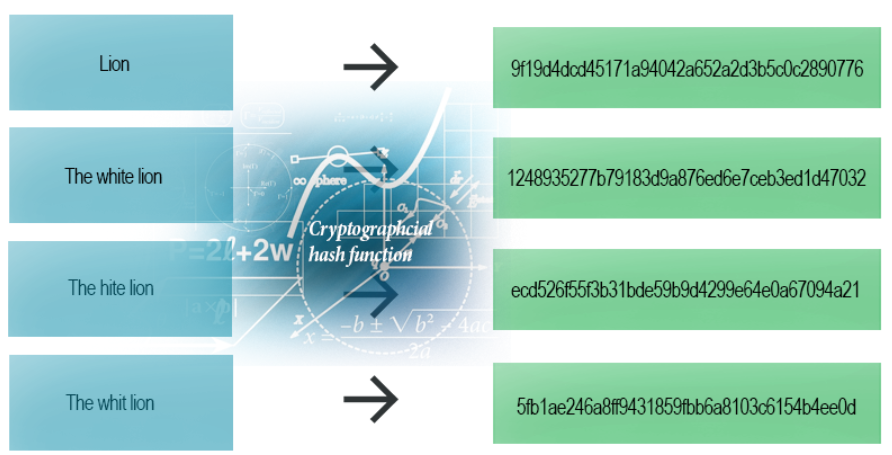
3.The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.



Diagrams, shown as a class diagram which is divided into 3 parts as mentioned above

**Methodology**

Firebase Authentication uses an internally modified version of scrypt to hash account passwords. Even when an account is uploaded with a password using a different algorithm, Firebase Auth will rehash the password the first time that account successfully logs in. Accounts downloaded from Firebase Authentication will only ever contain a password hash if one for this version of scrypt is available, or contain an empty password hash otherwise.



Above image shows how hashed text looks like.

**Modules with Implementation**

Implementing firebase complete module which is hosted and supported by google. Firebase provides the tools and infrastructure you need to develop, grow, and earn money from your app. This package supports web (browser), mobile-web, and server (Node.js) clients.

Firebase Realtime Database - The Firebase Realtime Database lets you store and query user data, and makes it available between users in realtime. The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our Android, iOS, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data. The Realtime Database provides a declarative rules language that allows you to define how your data should be structured, how it should be indexed, and when your data can be read from and written to. By default, read and write access to your database is restricted so only authenticated users can read or write data.

Firebase Storage - Firebase Storage lets you upload and store user generated content, such as files, and images. Cloud Storage for Firebase provides a declarative rules language that allows you to define how your data should be structured, how it should be indexed, and when your data can be read from and written to. By default, read and write access to Storage is restricted so only authenticated users can read or write data

Firebase Authentication - Firebase helps you authenticate and manage users who access your application. Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more. Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

**Expected Outcomes**

Our fully functional, stand alone website which uses real time database with no latency along with security which we all can trust on as firebase as service is provided by Google.