Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preamble

Technology driven solutions and analytics based decision making is the core of any business in the current scenario. Moreover, creation and integration of backend and frontend for a medium sized application requires 18 weeks of time comprising of 55% time[4] in backend development and average cost ranging from $8000 to $50,000. Moreover in today's scenario, there is a business urge to include statistics and other analysis to make crucial profit decisions. Growth of Analytics just as a Service is expected to be 23.4 bn dollars by 2021, and that of Backend as a service will be 208.1 bn dollars. Thenceforth, there arises the need for providing the developers with an automated tool with integrated features of backend and analytics development which will ultimately act as a profit making job in terms of cost, time and other crucial factors.

1.2 Need of the project

Developing the backend and analytics services at a single platform along with the minimization of the actual coding hours is the need of the hour; thereby increasing the throughput. At the same time, due to the digitization and the demand of analytics in business sector, there is a vast opportunity for the application developers to integrate analytics solution in their applications.

But for this, the developers have to perform tedious work which usually includes a lot of redundant and unproductive jobs (including the CRUD[1] applications, etc.) eventually in-

[1]CRUD - Create, Retrieve, Update and Delete operations performed on databases

creasing the development time and cutting down on various other important tasks like the front end development, rigorous testing, performing critical analysis, etc.

Using automated tools for performing and maintaining CRUD operations would ease the developer's task as it will reduce the latency caused by redundant or other housekeeping jobs like authentications/authorisation, third-party logins, etc.

Also, considering the analytics, the developers have to spend a lot of time studying and working on the algorithms that seems to be feasible to the situation. This also reduces the scope of algorithm comparison for the feasibility and optimality as the developer has to spend a lot of time creating different algorithms and then performing analytics with their help; so usually they come up with a theoretically optimal algorithm than a practical and tested/compared optimal ones.

1.3 Problem Statement

Software solution providers face constant challenge to provide a cutting edge solution to their clients along with the minimization of cost and efforts. They tackle problems related to compatibility of various technologies and constant maintenance of the software. This has been done for a long time using conventional approach of complete software coding along with a dedicated maintenance cycle. The approach has been successful but their are a lot of redundant tasks which span across projects and the goal is to minimize these tasks which can be done with minimal or no coding.

1.4 Objectives

The aim of the project is -

- **Stops unnecessary stack development:** Instead of many developers being forced to recreate a stack for each mobile app they develop, a BAaS service can provide for much of their underlying processing needs. The main issue would then be connecting to an API; instead of spending hours developing customized stacks, that have to be re-created, changed, and reassembled to fit the needs of each of the different app platform. Developers can build just what they need on top of the existing structures, instead of starting from scratch each time.

- **Allows for more accessibility:** If each app has the same underlying base, then BAaS has the potential to easily link apps across platforms. This has many benefits: from easier data sharing to better accessibility for cloud storage, a quicker spin-up time, and an overall better user experience.

- **Provides diverse outcomes from one model:** Think of BAaS like a ”starter home.” Each user starts off with the same basic elements and continues to add over those elements to create their own customized ”home.” However, the base elements of the house are all the same, other users have the potential to more easily understand and even interact with or fix the ”house,” creating a unified backend that has a better and stronger user base. Therefore, we are reducing the redundancy of writing the same code over applications.

- **Provide backend ready applications:** BAaS aims to provide the fastest way to build, deploy and manage a production application backend. To achieve this, BAaS provides a standardized architecture that includes as much automation as possible.

- **Provide a one-stop platform for various analytics:** BAaS aims at providing a platform for directly implementing the various types of machine learning and data analytics algorithms and thus enables a comparative study of algorithms on same situation and so the user(developer) could eventually choosing the most optimal as per his considerations.

1.5 Solution Approach

Various models are available for the development of a fully-fledged application as a developer. However taking the help of this project, the developer can have all the required functionalities and features as mentioned earlier.

As an approach to solving the problem, the developer will be provided with a software package which can be installed on the server. Various tools will be integrated and various APIs will be generated as the first step which will include the most necessary ones which are even crucial for a naive application like data APIs, auth APIs, logging and monitoring APIs and control APIs. As a second step, various other APIs can be added by the developer which will facilitate in terms of providing fully functioning separate modules which can directly be used as end-points for their application. The developer's convenience to use these APIs will also be taken into consideration as in the form of providing simple and easy to use User Experience for the same. As a parallel step, various analytical algorithms will also be developed to be provided to the developers as an on-the-go APIs/code. As a first step, we plan to integrate the most crucial 6-7 algorithms like of Classification, Clustering, Regression etc. The user can can select from a number of algorithms available to make a fully functioning funnel [2] whose entry point will have the live data stream and output being the final/live feed or report

1.6 Organization of the Project Report

The report has five chapters. Chapter 1 provides introduction about the project, its need, problem statement, objectives and solution approach. Chapter 2 is focused on the background of project which includes the area, available tools, hardware and software, the references used in the projects like the research papers and their limitations. Chapter 3 includes detailed problem statement, requirement analysis, functional and nonfunctional requirements, feasibility study, and other required diagrams which better present the picture.

[2]Funnel is a sequence of algorithms which operate one after another and the output of one algorithm is input to the successive one; thereby generating desired output at the end.

Chapter 4 includes the architectural diagrams, with the various design issues, architecture used, along with the insights of the various models, algorithms / approaches used. Chapter 5 presents the conclusion of the report with specifying the various phase wise implementation of the project.

Chapter 2

Background

2.1 Backend as a service(BaaS)

Developing a mobile application from scratch is more complicated task for developers because developers couldn't build features fast enough to keep up with market need.
First you have to research and select a technology stack, write the code, connect to 1 or more 3rd party cloud APIs, test your code, deploy the stack, make it secure, and constantly version and maintain everything when you add new features or 3rd party APIs change.

So the industry needed a platform to help developers manage the complex balance between need and supply. This need gave birth to BaaS.

Backend as a service (BaaS) is a cloud computing architecture that provides mobile applications with access to servers, storage, databases and other resources that they need to run on – quickly and effortlessly. BaaS is disrupting the traditional 'Mobile Enterprise Application Platform' for today's businesses by offering more turn-key functionalities than conventional API management to create better user experience (UX).

2.2 Analytics as a Service

For years, IT organizations have generated, collected and stored vast amounts of data. Now, IT is being asked not just to store the data but to provide the infrastructure to perform analytics on it. The trouble is that the task is a resource-intensive proposition.
According to Barret[8], Analytics as a service refers to the provision of analytics software and operations through web-delivered technologies. These types of solutions offer businesses an alternative to developing internal hardware setups just to perform business analyt-

ics.

2.2.1 Algorithms to be used in Analytics

At the start, 5 algorithms will be given as option to the user to integrate them in analytics. Other algorithms can then be integrated afterwards. These are:

1. *Neural Network*

   C. Stergiou and D. Siganos [8] defined Neural Network as ”An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information”.

   Application of Neural network:

   The neural network can be used in wide variety of classification,prediction and forecasting tasks such as:

   - Sales forecasting
   - Industrial process control
   - Customer research
   - Data validation
   - Risk management
   - Target marketing
   - Modelling and Diagnosing the Cardiovascular System
   - Credit Evaluation

2. *Linear Regression*

   Linear regression is a linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. Several linear regression analyses available are:

   - Simple linear regression: 1 dependent variable (interval or ratio), 1 independent variable (interval or ratio or dichotomous)
   - Multiple linear regression: 1 dependent variable (interval or ratio) , 2+ independent variables (interval or ratio or dichotomous)

- Logistic regression: 1 dependent variable (binary), 2+ independent variable(s) (interval or ratio or dichotomous)

Application of Linear Regression:

- Trend estimation
- In Economics: consumption spending, fixed investment spending, inventory investment, purchases of a country's exports, spending on imports, the demand to hold liquid assets, labor demand, and labor supply.
- In Finance: capital asset pricing model, risk analysis of investment
- In Business: sales of a product, pricing, performance

3. *K-Means Clustering*

   k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

   Application of K-Means Clustering:

   - Vector quantization
   - Cluster analysis
   - Feature learning

4. *Naive Bayes Classifier*

   Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

5. *Support Vector Machines*

   In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier

   Application of Support Vector Machine:

   - Text and hypertext categorization
   - Classification of images
   - Hand-written character recognition

6. *Naive Bayes Classifier*

   Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

2.3 Available Tools and Comparision

Table 2.1: *Feature Comparision between Existing tools*

Features	Parse	Firebase	Algolia	BAaS
Open Source	✓	✗	✗	✓
Data Storage	✓	✓	✓	✓
Real Time	✗	✓	✗	✓
Files	✓	✗	✗	✓
Authentication	✓	✓	✗	✓
Cloud Deployment	✗	✗	✗	✓
Push Notification	✓	✗	✗	✓
Data Analytics	✗	✗	✗	✓
Log Analytics	✗	✓	✗	✓

2.4 Hardware tools

- A dedicated server with minimum 1GB of RAM

2.5 Software tools

- NodeJS
- Python
- MongoDB

2.6 Literature review

BaaS has been a recent trend in the application development industry.It aims to boost up the development process,thereby saving time and cost.The literature review will focus on the keywords of the problem,its pros and cons.The review is majorly based on work done by Phy Nguyen[7].

2.6.1 Backend as a Service (BaaS)

In this section,pros and cons of BaaS will be discussed.

Pros

- More focus on UI/UX.
- More extensible with less efforts.
- Cross platform apps with one common backend.
- Development and Analytics at one place.
- Easy visualisation and Real time analysis
- Customisable analytics options

Cons

When it comes to disadvantages of a service, these are the categories : control, scalability, and shutting-down probability.

- Firstly, when an organisation has an application running over a BaaS, there is no complete control over its infrastructure and software stack (APIs, SDKs).
- Secondly, features of BaaS are not suitable for large-scale applications.
- Lastly, the risk that an organisation takes when they use BaaS, is that the service might get taken down and they will face a severe amount of data migration to a new backend system.

2.6.2 Analytics as a service

Pros

- Allows users to focus on exploring and analyzing data
- Teams can spend their time exploring the data, formulating hypotheses and discussing insights with collaborators.

Cons

- Might be a risky step.
- Cost and time required to transfer data to the AaaS provider.

Chapter 3

Analysis

3.1 Detailed Problem Statement

Backend is considered to be backbone of any application regardless the platform it is made for, since it is the backend which deals with basic CRUD operations involved in an application and it also provide data persistency. But still the major focus from users perspective is on front end or user interface,but developers on standard, put only 40% of their effort for front-end part remaining in backend related task.

The amount of work involved in creating this backend technology is never a simple task, there is a demand for various types of applications which demand to be completed within the prescribed deadlines.

It takes almost 18 weeks to develop and publish a standard native mobile application, which include almost 10 weeks to built a back-end and around 8 weeks to build the front-end.

The backend task contains a set of hectic steps in traditional approach including the following steps-

- Envisioning - 2-4 weeks.
- Iterative Development and Testing - 6-8 weeks.
- Stabilization - 1-2 weeks.
- Release - 2 weeks.

So, Total time (traditional approach),

App Making Cost = (UI/UX Design Hours + iOS App Hours + Android App Hours + Back-end Server Hours) $\times$ Developer's Hourly Rate.

Moreover, in today's scenarios as growing data and digitalisation of systems, the data can be used in various analytics related application, which has even worse condition since even small analytics requires a lot of effort to build on to the top of already developed application.

Mobile computing is growing rapidly many applications are cross-platform, and to create cross platform application we need centralised database which will support CRUD for various platform.

Hence to provide Backend and Analytics as a service, which deals with to provide API based automated solution to backend and analytics in a single unit. It will target to automate simple backend and analytic related task and provide endpoint APIs for the same.

3.2 Requirement Analysis

3.2.1 Functional Requirement

1. *Create schema :* User will be provided with option to create a schema of database to have initial database structure.
   Input- Schema design as per the user's (developer's) need.
   Output- Database table structure with all the required dependencies as per the schema.

2. *Create Template :* Based on the schema now user can choose the template (json structure as query response) as to retrieved when he hit the API.
   Input- Template format, chosen structure.
   Output- Data will be provided in the chosen format.

3. *Generate Endpoint :* Based on the schema now user can choose the template (json structure as query response) as to retrieved when he hit the API. After creating his template user can generate endpoint either in public or user specific mode.
   Input- Choose to create endpoints and mode.
   Output- Display the successful endpoints in an organised and easy to access way.

4. *Edit Template :* User is provided with option to customize template as per his need hence further modification for same endpoint.
   Input- Choose an endpoint and make customizations (Input as a new customised template).
   Output- Updated the modified template to the user's area.

5. *Authenticate Endpoint :*If user has selected for private access as per some id or database schema column, he can authenticate that endpoint.
   Input- Choose private access for the desired schema and provide some security code (id).
   Output- Authenticated schema (accessible only by valid authority).

6. *Generate tokens :* For authentication purpose a private and public tokens combination is generated.
   Input- Get user's credentials.
   Output- Generate unique authentication id and update/store it to database.

7. *Match tokens :* If user has selected for private access as per some id or database schema column, he can authenticate that endpoint.
   Input- Request API access with unique Id.
   Output- Provided authorization to that API.

8. *Use Custom Queries :* User will be provided with flexibility to write his own queries in case if he need more hold on the retrieved results.
   Input- Updated query with selected API and user's Id.
   Output- Query updated for particular user area and results modified.

9. *Show stats :* Stats such as number of active user, total API hits etc will be shown on the dashboard.
   Input- Visit Dashboard update (auto refresh).
   Output- Current status of active users, API hits, etc.

10. *CRUD Operation :* Operations such as Create, read, update and delete in background if user hit the API endpoint.
    Input- API endpoint, with user id; data, and action to work on.
    Output- Update the database and notify the user.

11. *Logging Info :* Informations related to exception, Access etc are logged for future references.
    Input- Real time website data.
    Output- Logging details i.e. based on IP, page clicks, exception or error etc.

12. *Export Data :* Export data stored in schema or other logs related to statistics can be dump into various formats.
    Input- Select schema or info to export and file format.
    Output- Export data in standard file format eg csv, xml, json etc.

13. *Import Data :* Options Available to import the data from some predefined file format.
    Input- Select import file and its file format.
    Output- Data imported according to file format in schema.

14. *Create Funnel :* Selecting a sequence of algorithms for the desired analysis tasks and saving it for multiple use.
    Input- Select the algorithms, the data inputs along with the specified parameters and the format of the result.
    Output- Endpoint which give result of analytics algorithm in chosen schema.

15. *Choose Labels :* Select features from specific schema to apply algorithm/funnel.

16. *Generate Graphs :* Generate graph on the basis of insights generated from the analysis.
    Input- Select attributes from the set of all relevant attributes, for plotting the graph.
    Output- Generated graph corresponding to the data on specified attributes.

17. *Generate Summary :* Small summary based on the analytics results.
    Input- Select the data-set to be summarised.
    Output- Brief summary related to the trends or any anomaly, etc.

18. *Aggregation functions :* Availability of default aggregation function such as Min, Max, Mean, Mode etc for direct usage.
    Input- Calls to the predefined (custom-made) functions, and parameters to work upon.
    Output- Output the results related to the desired function.

19. *Social Media Login/Logout :* Social media integration such as login via facebook, google+, twitter etc.
    Input- Choosing the third-party, application ID and secret key.
    Output- Third-party login service will be embedded.

20. *File storage :* API endpoints specifically for data files in case if files are stored.
    Input- Choose for file handling operations.
    Output- Output or make the desired changes.

3.2.2 Non-Functional Requirement

1. *Security :* The security in every aspect will be taken into consideration by various authentication and authorization processes at every crucial place like API access, data entry, customization, etc.

2. *Reliability :* The APIs will be created as a full and tested ready-to-use code which will ensure a reliable services.

3. *Flexibility :* Change and modification in functionalities is easier. For a large scale application, modification in features without affecting other functionalities is desired.

4. *Reusability :* Provides a vast reusable code in the form of APIs, algorithms, etc. Also the customization in the APIs could also be used further so the customized query and other factors.

5. *Maintainability :* The control over the application is an aid as the modules can be easily tested for bugs or enhancements which are usually loosely coupled. So a module wise implementation ensures individual maintainability as well.

6. *Extensibility :* Adding new features and functionalities to an existing system becomes easier and faster with already available APIs.

7. *Auditability :* With the logging and monitoring tools, any anomaly and inoperable state are communicated in real time.

3.3 Feasibility Study

1. **Economical :** Languages tools and software which are used to develop the product are easily available for free.

2. **Technical :** Tools and software needed for realisation of the system such as NodeJS, Python, C etc are easily available and can be integrated.

3. **Operational :** Proposed system will be operationally very efficient it will guarantee to reduce total application development time by at least 30%+ which is a very great amount of time

Chapter 4

Design

The diagrams that are necessary to analyse, visualize and evaluate the system are discussed in this chapter. Architecture, use case and ER diagrams are created to know the complete design of the system.

4.1 Diagrams

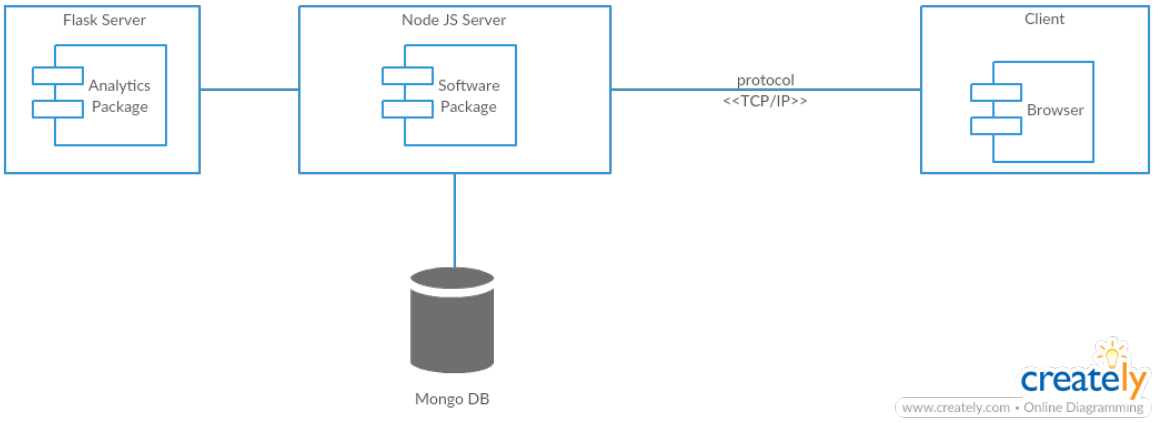

Figure 4.1: Deployment Diagram

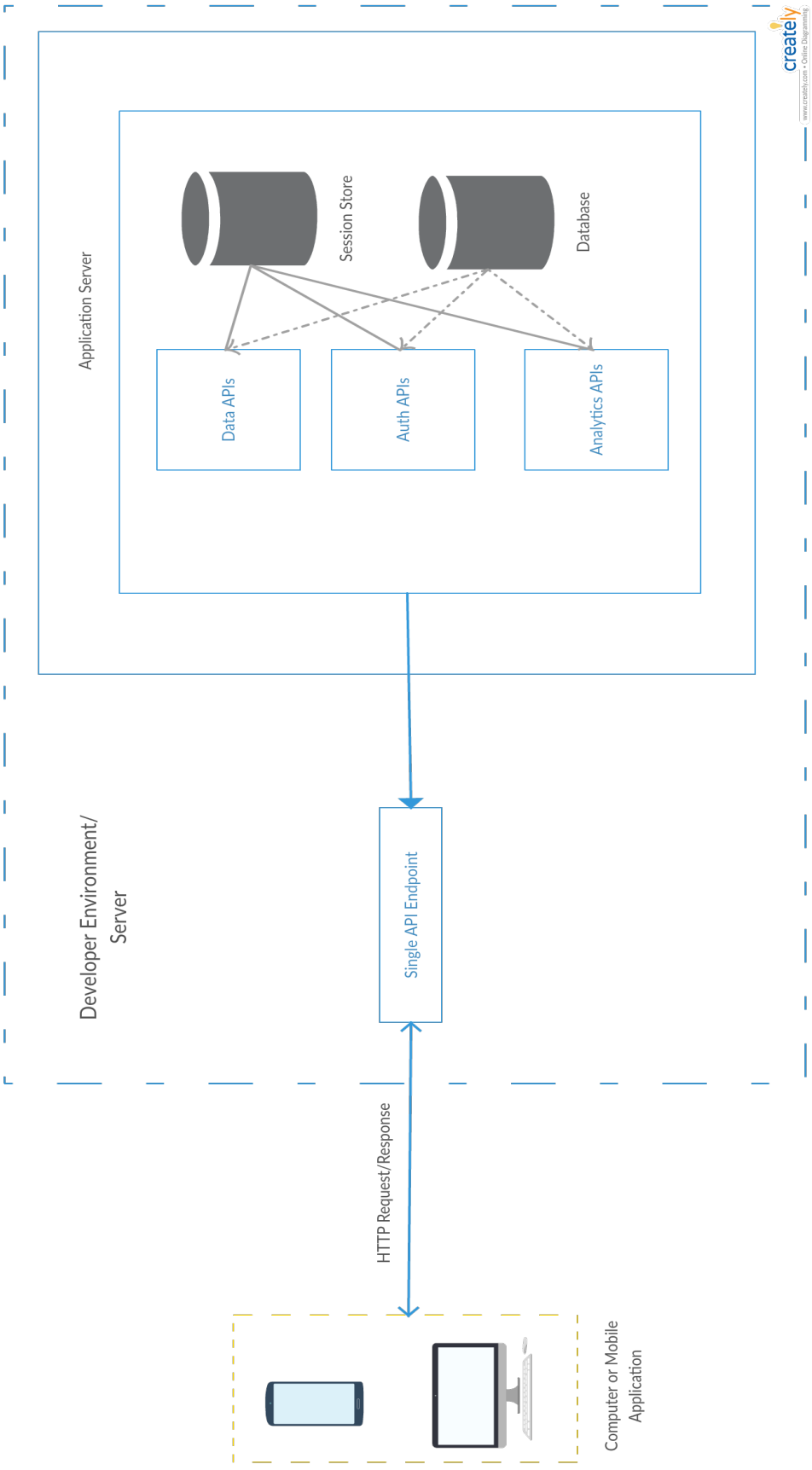

Figure 4.2: Architecture Diagram

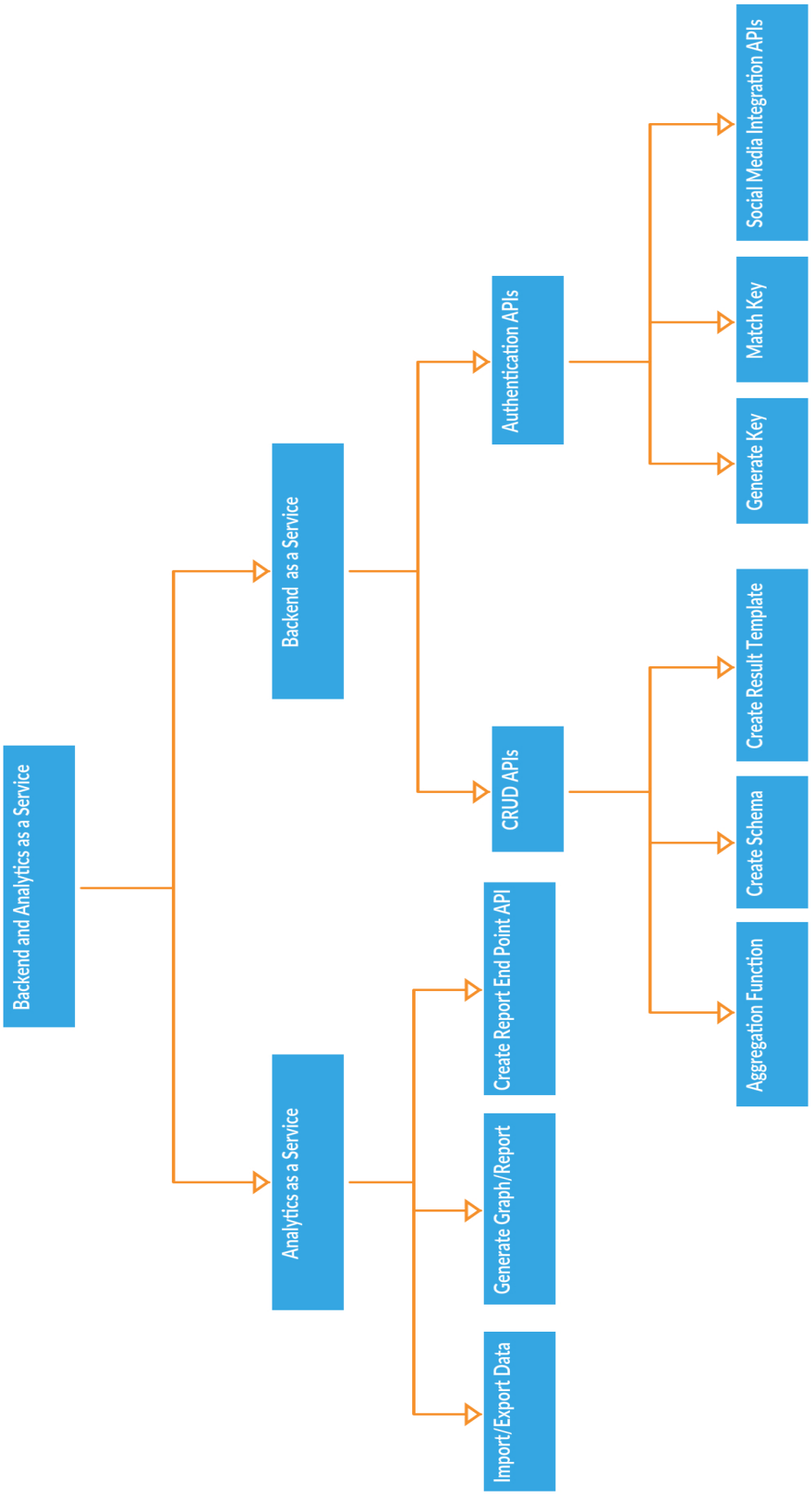

Figure 4.3: Hierarchical Diagram

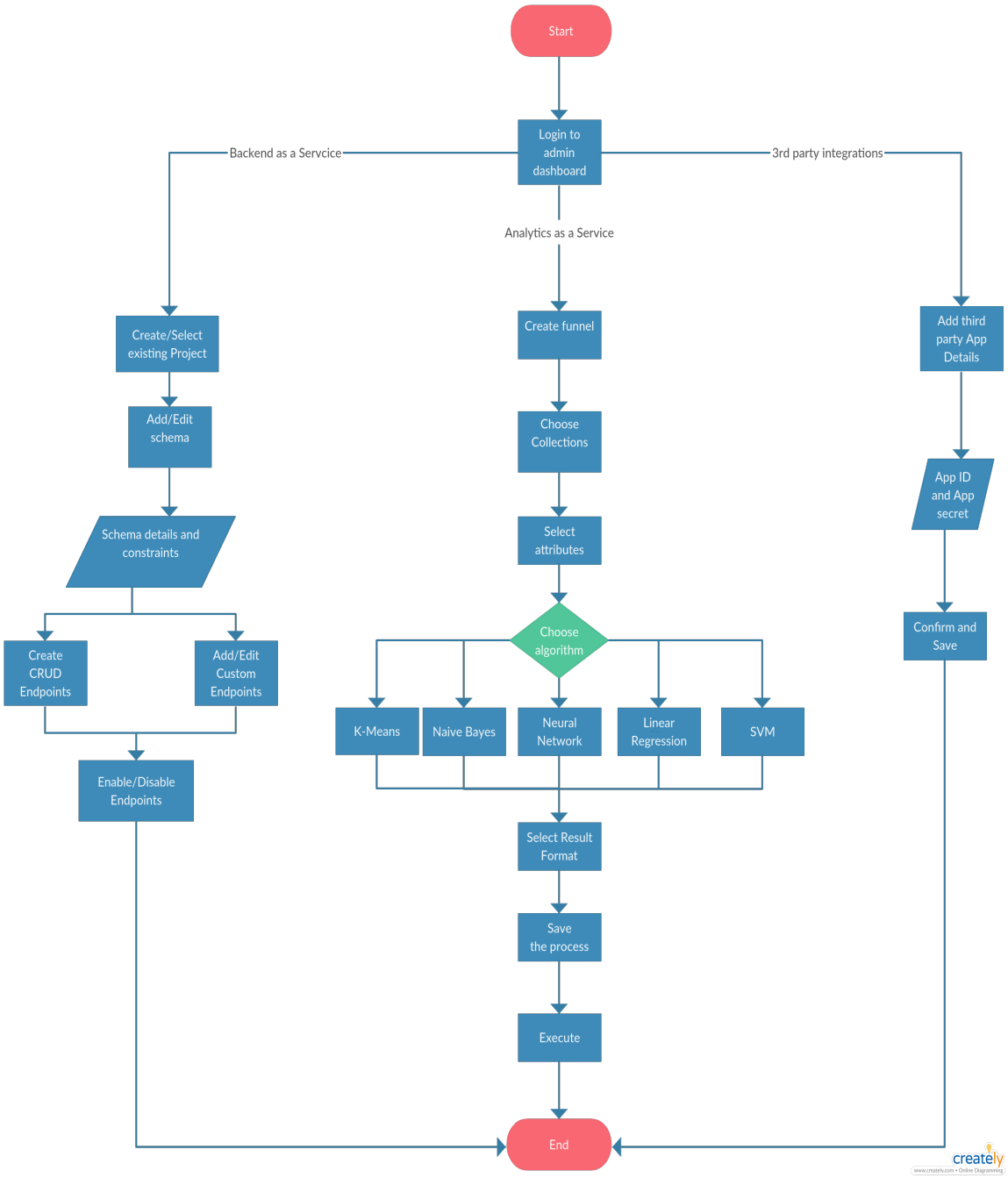

Figure 4.4: System Flowchart Diagram

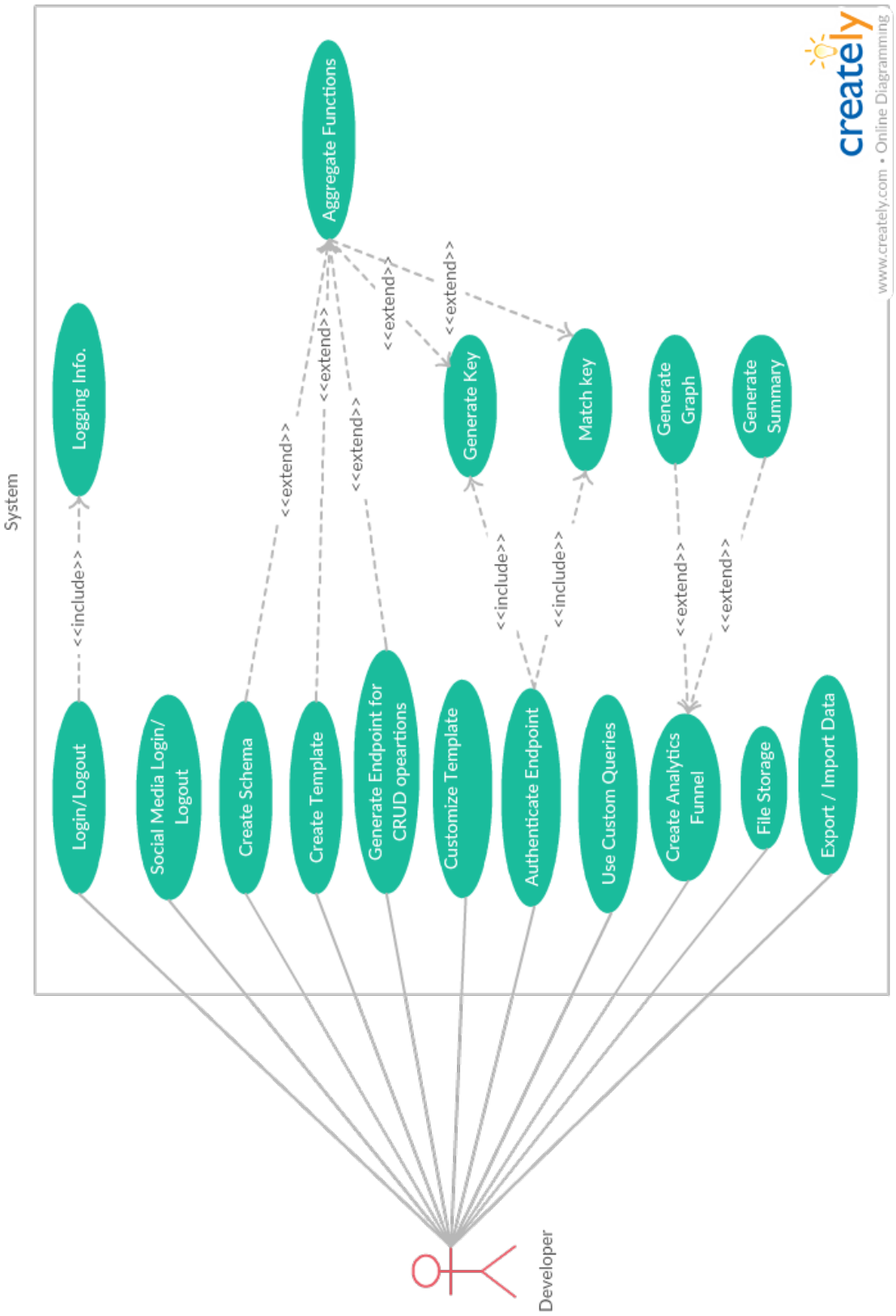

Figure 4.5: Usecase Diagram

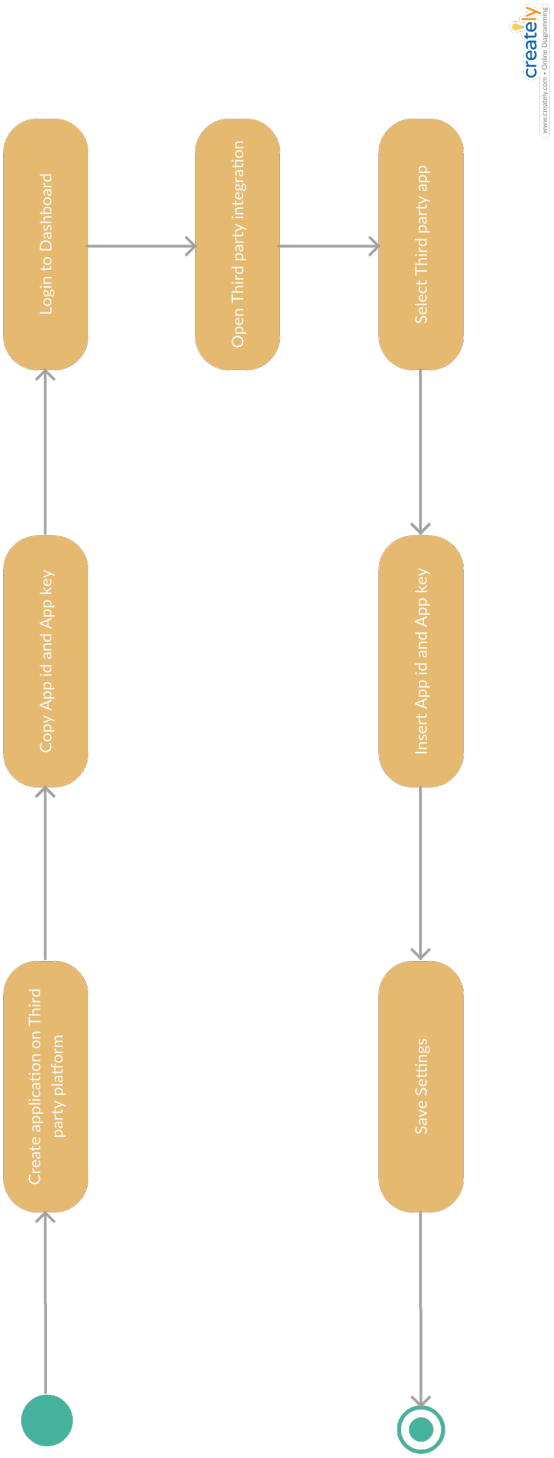

Figure 4.6: Activity Diagram - Social Login Integration

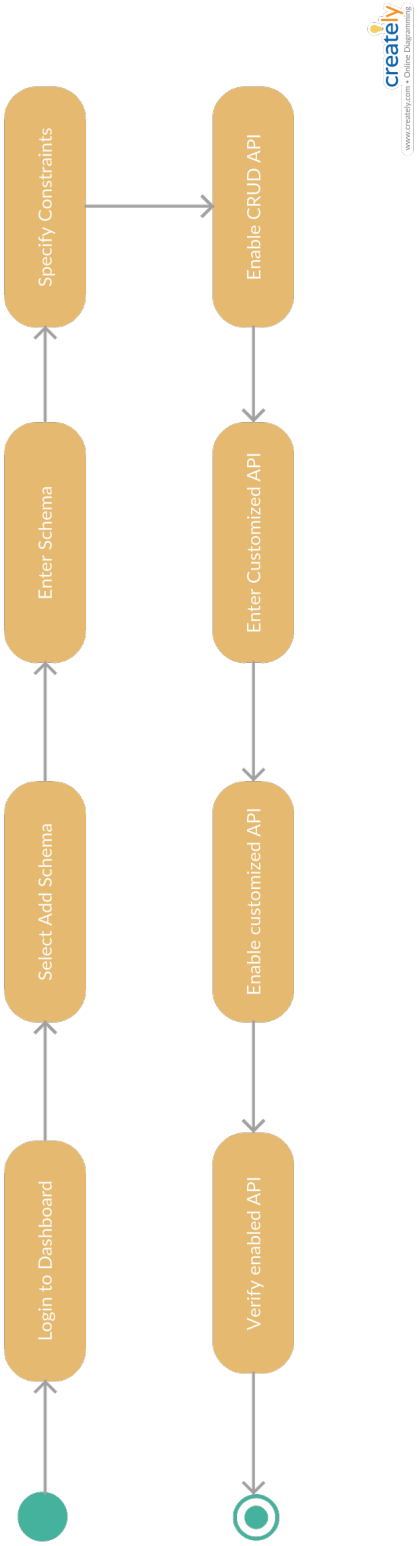

Figure 4.7: Activity Diagram - Generate Schema and CRUD APIs

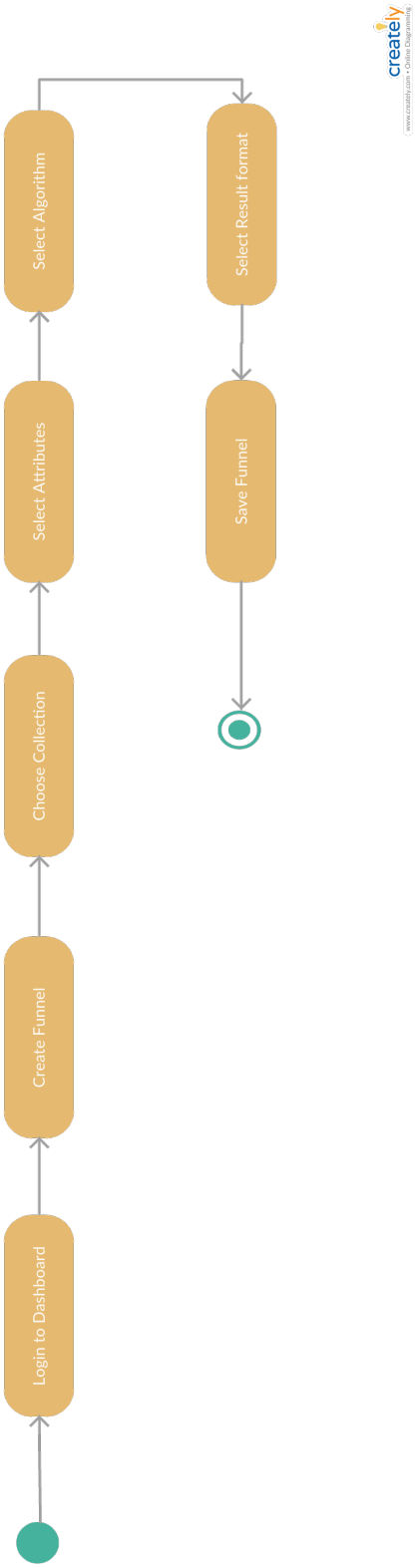

Figure 4.8: Activity Diagram - Analysis APIs Generation

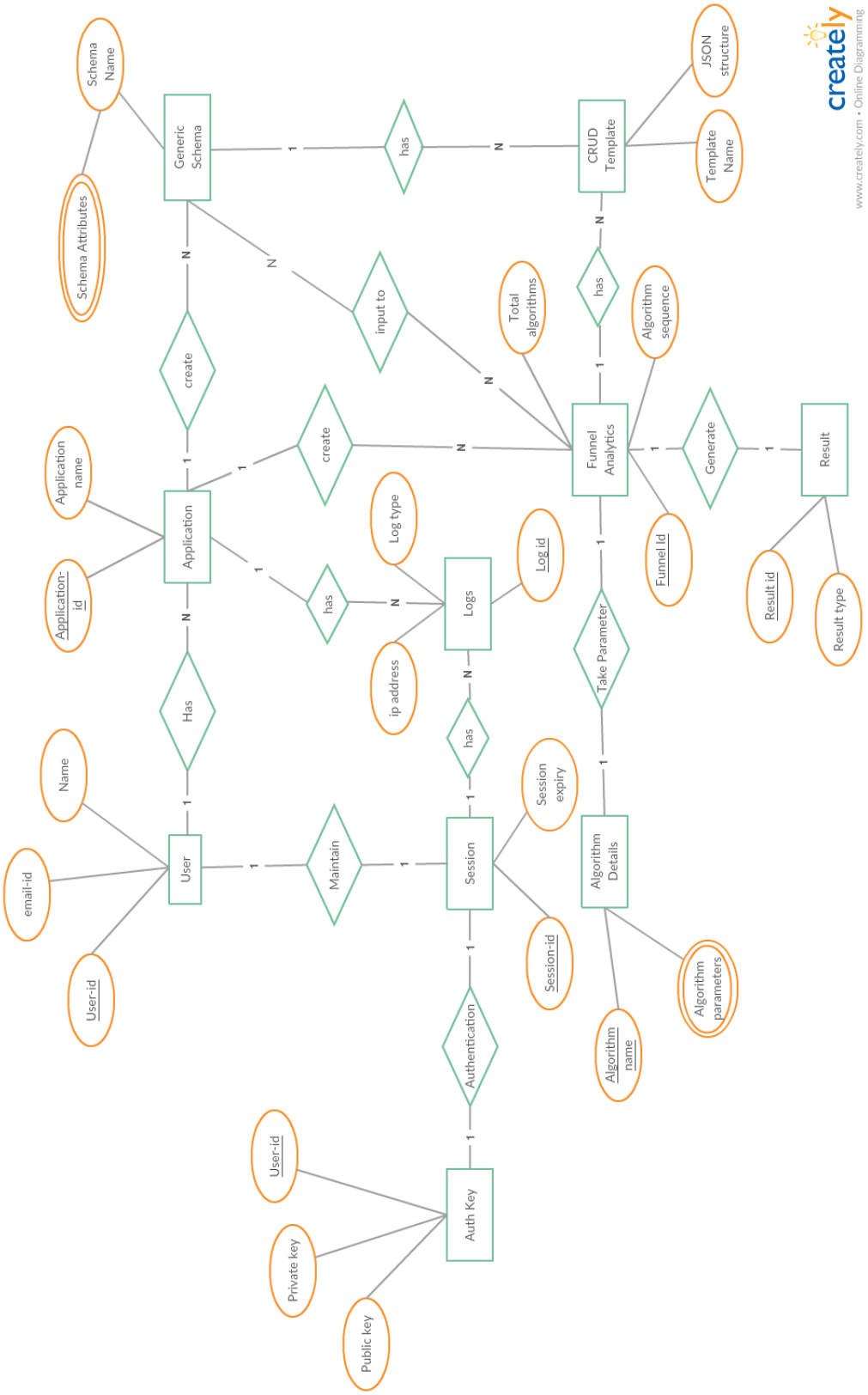

Figure 4.9: ER Diagram

Chapter 5

Conclusion

This chapter consists of the work carried out in phase I and the tasks to be carried out in phase II of the project.

5.1 Work carried out in phase 1

- Detailed study of Existing Systems
- SRS(Software Requirement Specification)
- Project Report and documentation
- Finalised functional and nonfunctional requirements and features
- Finalised our technology stack

5.2 Work to be carried out in phase 2

- Creating User Interface and Configuring server
- Implementation of the software package
- Creating Endpoints for the algorithms
- Implementation of analytical tools

References

[1] Francis Gropengießer and Kai-Uwe Sattler, *Database Backend as a Service: Automatic Generation, Deployment, and Management of Database Backends for Mobile Applications*. Available at https://link.springer.com/article/10.1007/s13222-014-0157-y

[2] *Overview Of Backend as a Service (BaaS) White Paper*. Available at http://baas.apievangelist.com/2013/05/08/overview-of-backend-as-a-service-baas-white-paper/

[3] Balkan, A. 2012, *What are the pros and cons of using a backend as a service?*. Available at https://www.quora.com/What-are-the-pros-and-cons-of-using-a-backend-as-a-service

[4] Backend as a Service (BaaS) Market Global forecast for 2020, Available at http://www.marketsandmarkets.com/PressReleases/baas.asp

[5] Backendless website 2016, *Database Backend as a Service: Automatic Generation, Deployment, and Management of Database Backends for Mobile Applications*. Available at https://backendless.com/what-is-backend-as-a-service/

[6] Ryan Goodrich, *What is BaaS (Backend as a Service)*. Available at http://www.businessnewsdaily.com/4992-what-is-baas.html

[7] Nguyen, T, 2016, *Mobile Backend as a Service: The Pros and Cons of Parse Case company: SuperApp Oy*. Available at http://theseus.fi/bitstream/handle/10024/117483/Nguyen_Phu.pdf?sequence=2

[8] C. Stergiou and D. Siganos , 2017, *Neural Network*. Available at https://www.doc.ic.ac.uk/ nd/surprise_96/journal/vol4/cs11/report.html

[9] *What is Linear Regression.* Available at http://www.statisticssolutions.com/what-is-linear-regression/