PROJECT IMPLEMENTATION

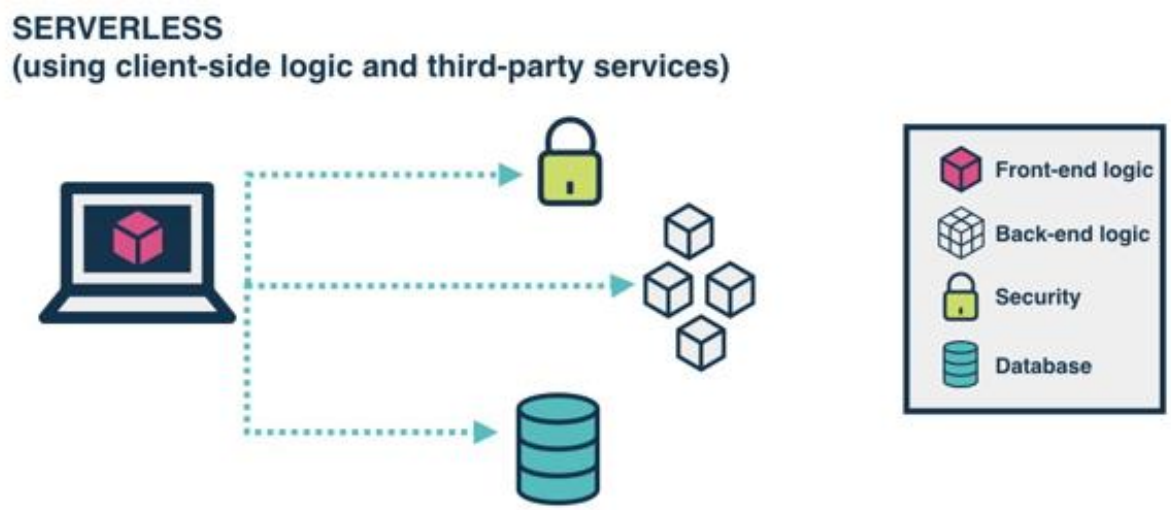
Project Name: Save Planet Earth **Date:** 01/03/19

Group Members of ACA Engineers:
- Anissa Harricharan: 816008114
- Celine Ganar: 816008305
- Aakil Ramlogan: 816007871

Architectural Design

Component diagrams, high level descriptions of the components in the system and their purpose in relation to the project's objectives.

Serverless Architecture saves a lot of time taking care and fixing bugs of deployment and servers regular tasks. Serverless architecture (also known as function as a service, FaaS) is a software design pattern where applications are hosted by a third-party service, eliminating the need for server software and hardware management by the developer.



We are using Contentful in our application, a serverless architecture, which makes it as easy as making an API call. Using a serverless architecture means that it will not have problems dealing with provisioning and maintaining servers, even when production peaks are hit. There is focus on content, rather than servers and maintenance with this Serverless Architecture, Contentful. The benefits include:

- Focuses on coding and producing content, not maintaining backend databases.
- Boosts effectiveness and speed of cross-platform development using microservices.
- Easily scalable, flexible and customizable with features such as UI extensions.

Source website: https://www.contentful.com/r/knowledgebase/serverless-architecture/#serverless-architecture-by-example

Class Diagram

Outlines the attributes, methods and interactions of the major classes/modules in the system.

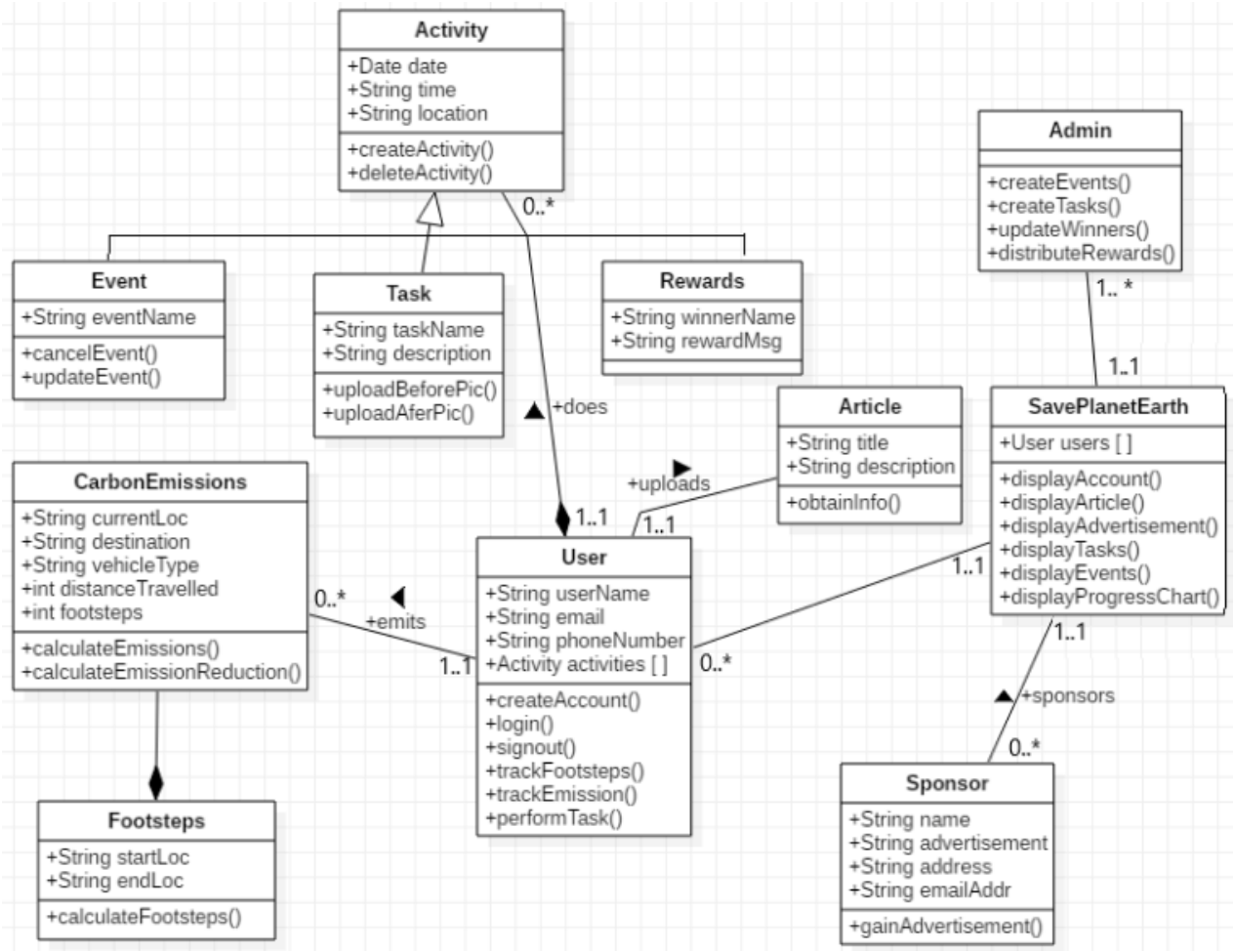


Figure 1.0: Showing the Class Diagram of the entire Save Planet Earth System.

Entity Relationship Model Diagram

Specifies the entities, datatypes and relationships that are important for the project domain.

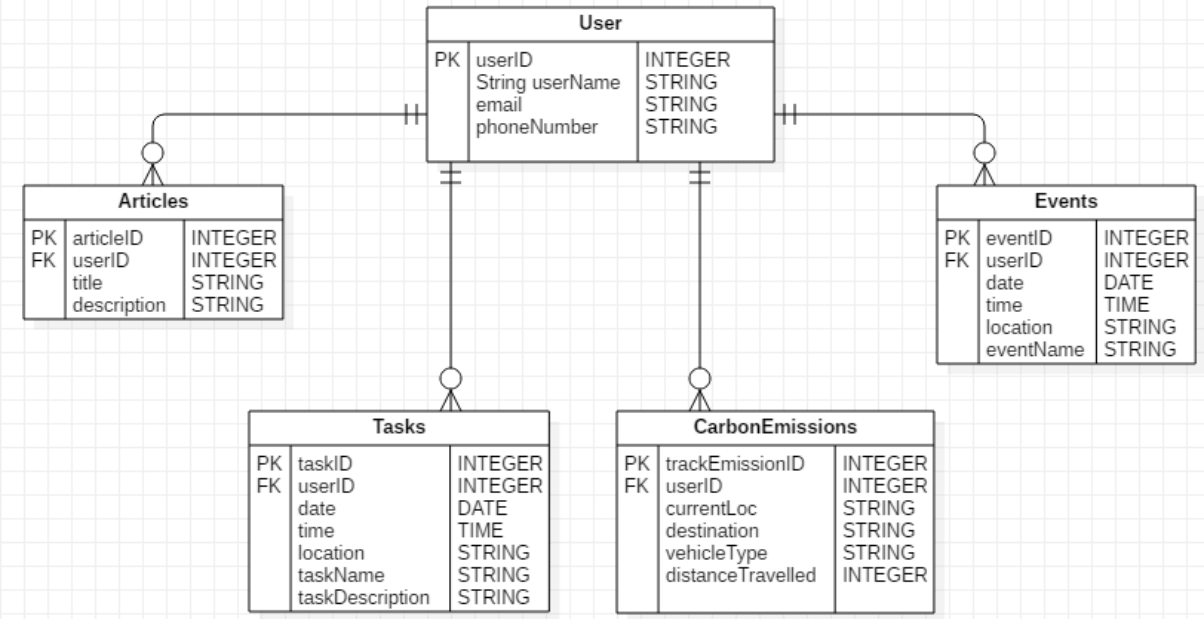


Figure 2.0: Showing the Entity Relationship Model Diagram of the major relationships of the Save Planet Earth System.

Sequence diagram:

Illustrates object interactions for the major use case scenarios for the project.

Logging In

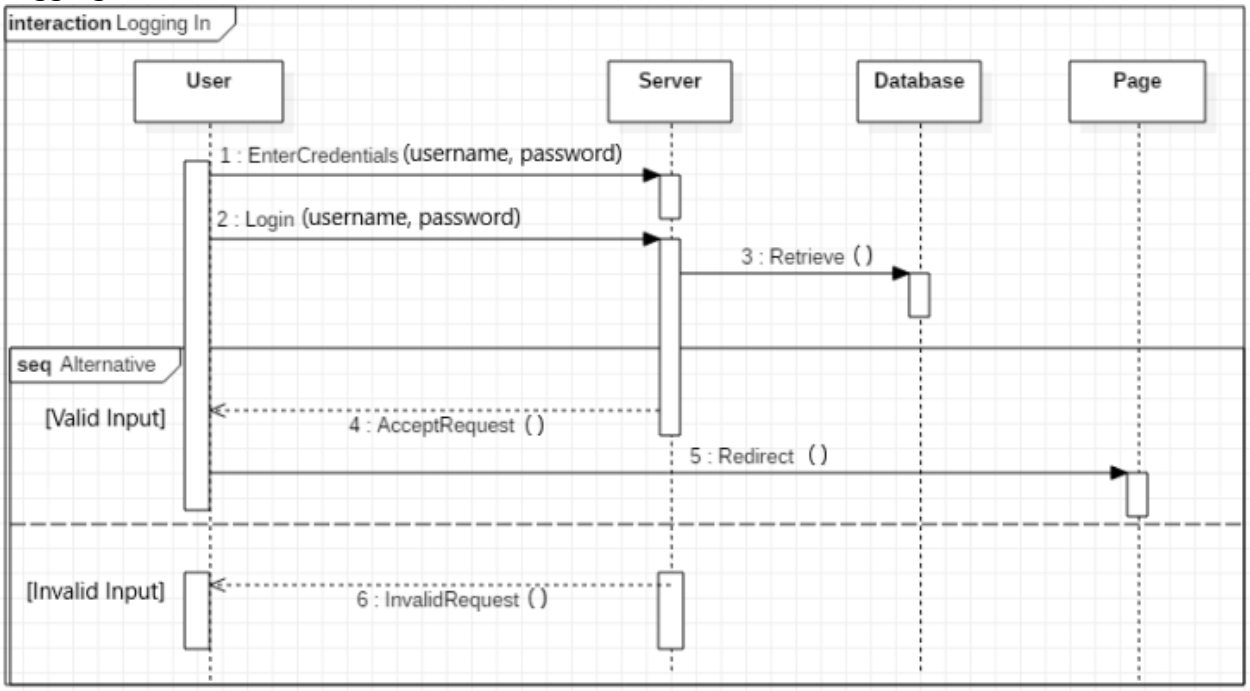


Figure 3.0: Showing the Sequence Diagram of Logging In of the Save Planet Earth System.

Obtaining information from Articles

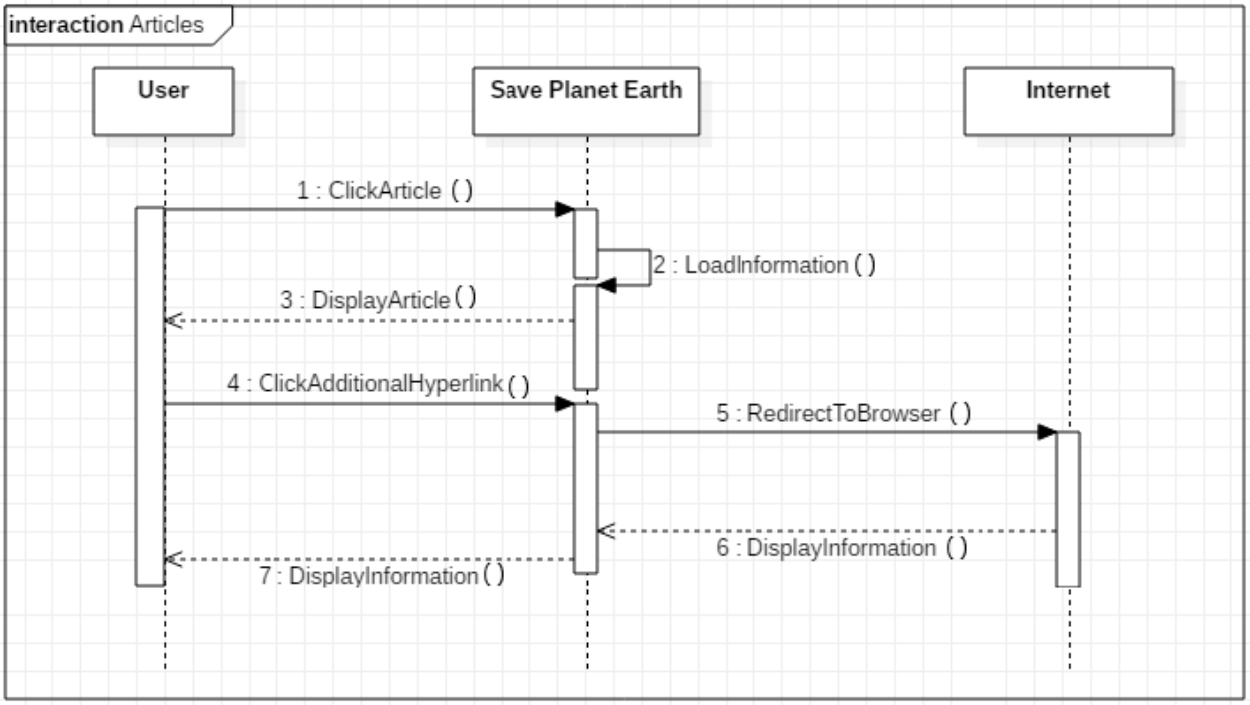


Figure 3.1: Showing the Sequence Diagram of Obtaining Environmental Conservation Information from Articles in the Save Planet Earth System.

Context Diagram

It identifies the flows of information between the system and external entities to clarify the boundaries of the software system.

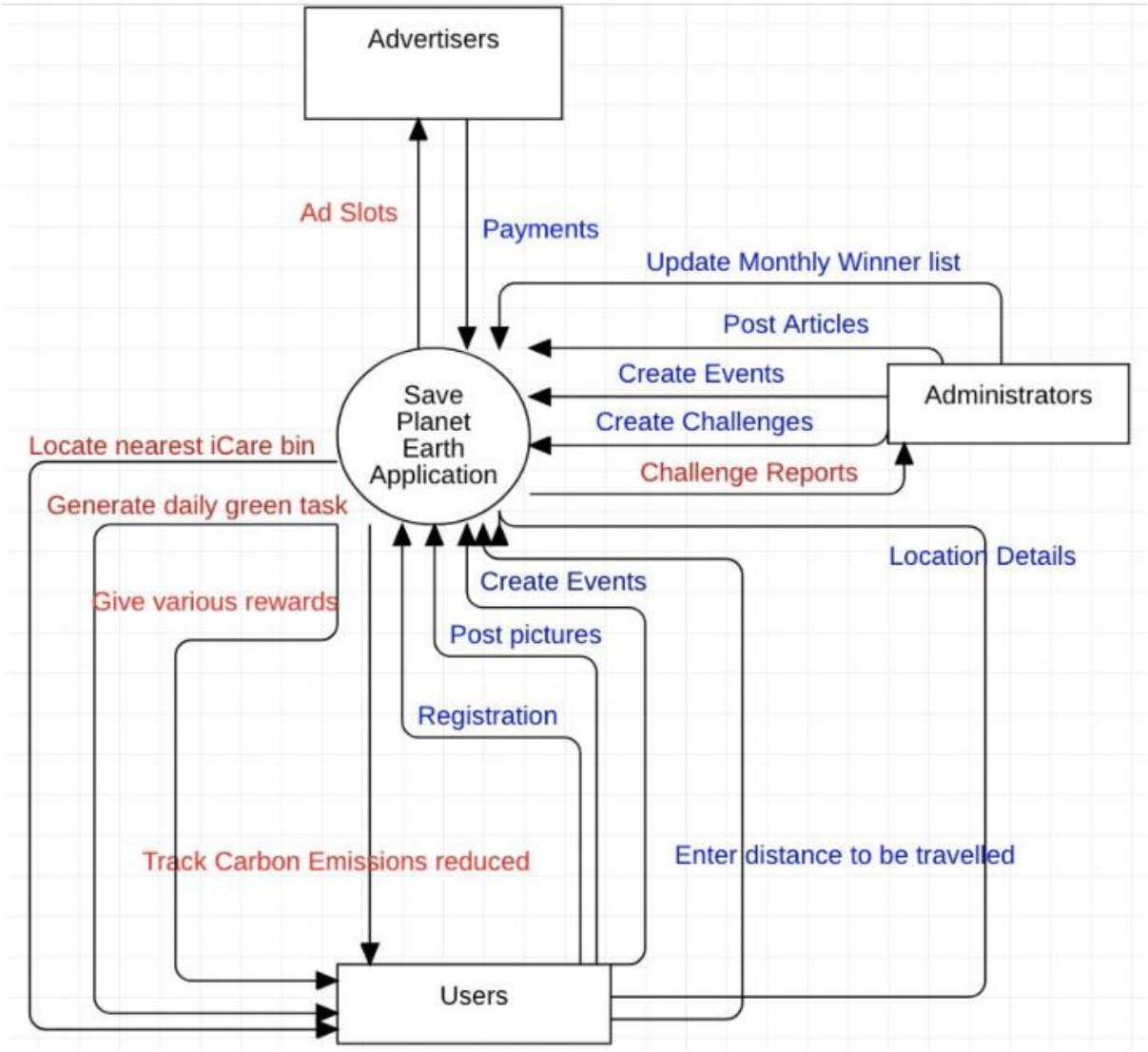


Figure 4.0: Showing the Context Diagram of the entire Save Planet Earth System.

Description of Technology used

Hardware devices, software products, programming languages.

The hardware devices used to build this cross-platform website consist of two Mac and one Windows PC as well as two Apple smartphones and one Android. The PCs were used for the actual coding of the website and the programming languages used are HTML, CSS and JavaScript. The smartphones were used for testing purposes and to ensure the website works successfully not only on a laptop, but on other devices as well. Other software includes the use of Contentful and Eventbrite in creating the Events page as well as frameworks such as Angular and Postman for the EventBrite API. Additionally, Google Firebase was utilized for the authentication of users while they register and login. Code was shared with all members of the development team via GitHub and to track the progress of the sprints. Lastly, StarUML was used to design the Class Diagram, Entity Relationship Model Diagram, Context Diagram and Sequence Diagrams required for this document.