

Online Event Scheduler Technical Design Document

1. Tech Stack

FRONTEND:

- HTML: HyperText Markup Language is the code that is used to structure a web page and its content.
- CSS: Cascading Style Sheets are used to style HTML Pages.
- ReactJS: Open source Javascript library used to build user interfaces. It is maintained by Meta. We make use of ReactJS to build end-user screens.
- ReactDOM: ReactDOM is the JavaScript library that allows React to interact with the Document Object Model(DOM). We use ReactDOM to manage the DOM elements of the web app.
- react-router-DOM: The react-router-dom package contains bindings for using React Router in web applications.
- Bootstrap: Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

BACKEND:

- Flask : Python web framework used for building web applications and APIs.
- Python: Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility in various applications.

DATABASE:

- MySQL: An open-source relational database management system used for storing, organizing, and retrieving data in various applications.

TESTING:

- Selenium: A popular open-source framework for automating web browsers and testing web applications.
- Postman: A popular tool used for testing APIs, making HTTP requests, and validating responses.

2. Accounts and Infrastructure

2.1 Development

- The team members will use their own personal machines to host a local development environment for the web application. This allows each member to work independently on their own codebase and test their changes in a local environment before committing them to GIT.
- The development/QA environment will be hosted on IBM Cloud, which allows developers to work on a shared codebase and test their changes in a shared environment. This can help to ensure that the code is compatible with other components of the application and that changes do not cause unintended side effects.
- Access to the development environment will typically require authentication and access control measures to ensure that only authorized users can access the system. Login credentials and URLs for the development environment will be handled by one team member and may vary depending on the specific development environment being used.

2.2 Production

- OES will be hosted in the cloud, more likely on IBM cloud sponsored by the CSE department.
- OES will be provisioned with various security measures to protect user data, including encryption and hashing of data in transit and at rest, authentication, authorization, and vulnerability assessments.
- Overall, the production environment for OES will be designed to handle a large number of concurrent users and ensure high availability and reliability.

Data Sources, Models, Timing

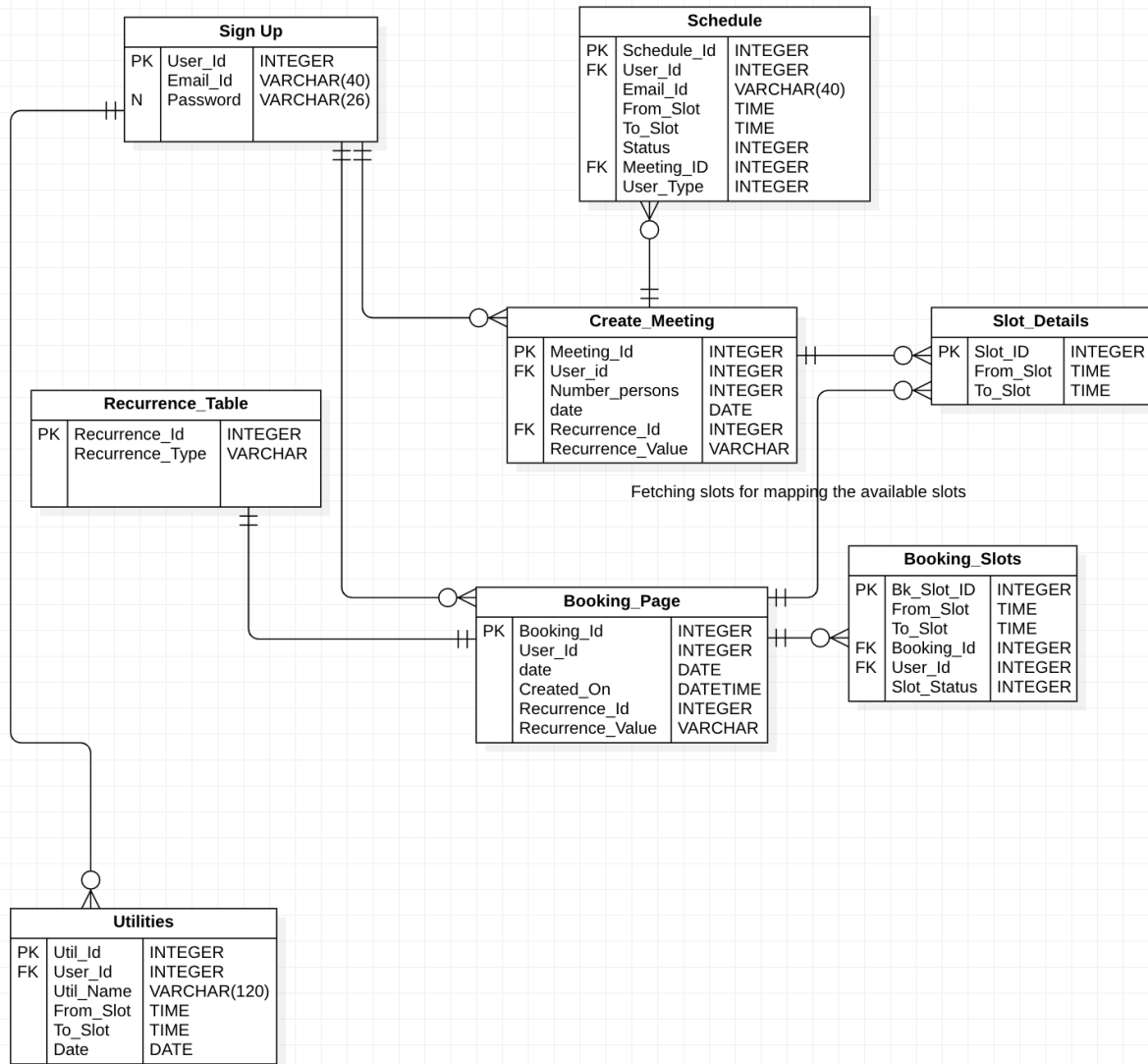
1.1 Data Sources

- The data for OES will likely come from a variety of sources, including user input and third-party data sources. For example, users will input their availability and scheduling preferences into the application, which will be used to generate a list of possible meeting

times. The application may also integrate with third-party calendar applications, such as Google Calendar or Outlook, to import existing scheduling data.

- In terms of data management, the OES application may use a combination of user-entered data and seeded data. Seeded data refers to data that is pre-populated into the application's database, either manually or through automated scripts. This data may include default settings, system preferences, or other configuration data that is necessary for the application to function correctly.
- Administrators will be responsible for configuring initial data and maintaining the application's database over time. This may involve tasks such as creating new user accounts, managing user permissions, and monitoring the application's performance and data integrity. Additionally, the application will use automated data management tools and techniques, such as data backups and restores, to ensure that data is protected and recoverable in the event of an outage or other system failure.

1.2 Data Models and Structure



Relations, Attributes, Primary key and Foreign Key:

1. Create_Meeting

Attributes and Datatypes: Meeting_Id (Integer), Number_persons (Integer), User_id (Integer), Recurrence_Id (Integer), Recurrence_value (varchar), date (Date)

Primary Key: Meeting_Id

Foreign Key: User_Id referenced from table Sign_Up, Recurrence_Id referenced from table Recurrence_Table.

Attribute Description:

Meeting_Id: This is the primary key of this relation and it is auto generated by the application and it is of type Integer.

Number_persons: This attribute denotes the maximum meeting room capacity and its of type Integer

User_id: This attribute stores the user_id of the person who created the event/poll and it is of type Integer.

Recurrence_Id: This attribute denotes the id of the recurrence type(month,week,year,day) and it is of type Integer.

Recurrence_value: This attribute stores the value based on the recurrence type and it is of type varchar.

Date: This attribute stores the value of date for which availability has been selected and it is of type date.

2. Schedule

Attributes and Datatypes: Schedule_Id (Integer), Email_Id (Varchar), User_Id (Integer), From_Slot (Time), To_Slot (time), Status (Integer), Meeting_Id (Integer), User_Type (Integer),

Primary Key: Schedule_Id

Foreign Key: User_Id referenced from table Sign_Up, Meeting_Id referenced from table Create_Meeting.

Attribute Description:

Schedule_Id: This is the primary key of this relation and it is auto generated by the application and it is of type integer.

Email_Id: This attribute stores the email_id of the users and it is of type varchar. The size of this attribute is limited to 40 characters.

User_Id: This attribute stores the user_id of the person who poll their availability and it is of type Integer.

From_Slot: This attribute stores the starting value of the selected range and it is of type time.

To_Slot: This attribute stores the ending value of the selected range and it is of type time.

Status: This attribute denotes the status of poll and it is of type integer.

Meeting_Id: This attribute identifies the participants and schedule information of the meeting

User_Type: This attribute denotes the type of user (initiator or participant) and it is of type integer.

3. Sign_Up

Attributes and Datatypes: User_Id (Integer), Email_Id (Varchar), Password (Varchar)

Primary Key: User_Id

Attribute Description:

User_Id: This attribute is the primary key of this relation and it is auto generated by the application and it is of type Integer.

Email_Id: This attribute stores the email of the user and it is of type varchar.

Password: This attribute stores the password of the user and it is of type varchar.

4. Slot_Details

Attributes and Datatypes: Slot_Id (Integer), From_Slot (Time), To_Slot (time),

Primary Key: Slot_Id

Attribute Description:

Slot_Id: This attribute is the primary key of this relation and it is auto generated by the application and it is of type Integer.

From_Slot: This attribute stores all the available slots (starting range based on the duration) based on the requirement and it is of type time.

To_Slot: This attribute stores all the available slots (ending range based on the duration) based on the requirement and it is of type time.

5. Recurrence_Table

Attributes and Datatypes: Recurrence_Id (Integer), Recurrence_Type (Varchar)

Primary Key: Recurrence_Id

Attribute Description:

Recurrence_Id: This attribute is the primary key of this relation and it is auto generated by the application and it is of type Integer.

Recurrence_Type: This attribute denotes the type of recurrence such as week, month, year and day and it is of type varchar.

6. Booking_Page

Attributes and Datatypes: Booking_Id (Integer), User_Id (Varchar), Created_On (Integer), Recurrence_Id (Integer), Recurrence_value (varchar), date (Date)

Primary Key: Booking_Id

Foreign Key: Recurrence_Id referenced from table Recurrence_Table

Attribute Description:

Booking_Id: This attribute is the primary key of this relation and it is auto generated by the application and it is of type Integer.

User_Id: This attribute stores the user_id of the user who is going to create the booking page and it is of type integer.

Created_On: This attribute stores the date and time of when the booking page has been created and it is of type date time.

This attribute denotes the id of the recurrence type (month, week, year, day) and it is of type Integer.

Recurrence_value: This attribute stores the value based on the recurrence type and it is of type varchar.

Date: This attribute stores the value of date for which availability has been selected and it is of type date.

7. Booking_Slots

Attributes and Datatypes: Bk_Slot_Id (Integer), From_Slot (Time), To_Slot (time), Booking_Id (Integer), User_Id (Integer), Slot_Status (Integer)

Primary Key: Booking_Id

Foreign Key: User_Id referenced from table Sign_Up, Booking_Id referenced from table Booking_Page

Attribute Description:

Bk_Slot_Id: This attribute is the primary key of this relation and it is auto generated by the application and it is of type Integer.

From_Slot: This attribute stores the starting value of the selected range and it is of type time.

To_Slot: This attribute stores the ending value of the selected range and it is of type time.

Booking_Id: This attribute identifies the participants and schedule information of the booking

User_Id: This attribute stores the user_id of the person who selected the slot and it is of type Integer.

Slot_Status: This attribute denotes the status of the booked slot and it is of type integer.

8. Utilities

Attributes and Datatypes: Util_Id (Integer), User_Id (Integer), User_Name (Varchar), From_Slot (Time), To_Slot (time), Booking_Id (Integer)

Primary Key: Util_Id

Foreign Key: User_Id referenced from table Sign_Up

Attribute Description:

Util_Id: This attribute is the primary key of this relation and it is auto generated by the application and it is of type Integer.

User_Id: This attribute refers to the user_id of the user and it is of type Integer.

Util_Name: This attribute stores the name of the util like to do tasks, task reminders and it is of type varchar.

From_Slot: This attribute stores the starting time of task and its of type time

To_Slot: This attribute stores the end time of task and its of type time

Booking_Id: This attribute stores the booking_id of the booking page and it is of type integer.

1.3 Timing

Timing and Method:

- The OES web application allows users to create and share scheduling grids with others via unique URLs.
- All scheduling data, including participant availability and event details, is entered directly into the web application by users via dynamic selection and stored in a database.
- There are no imports or exports of data in the OES web application.

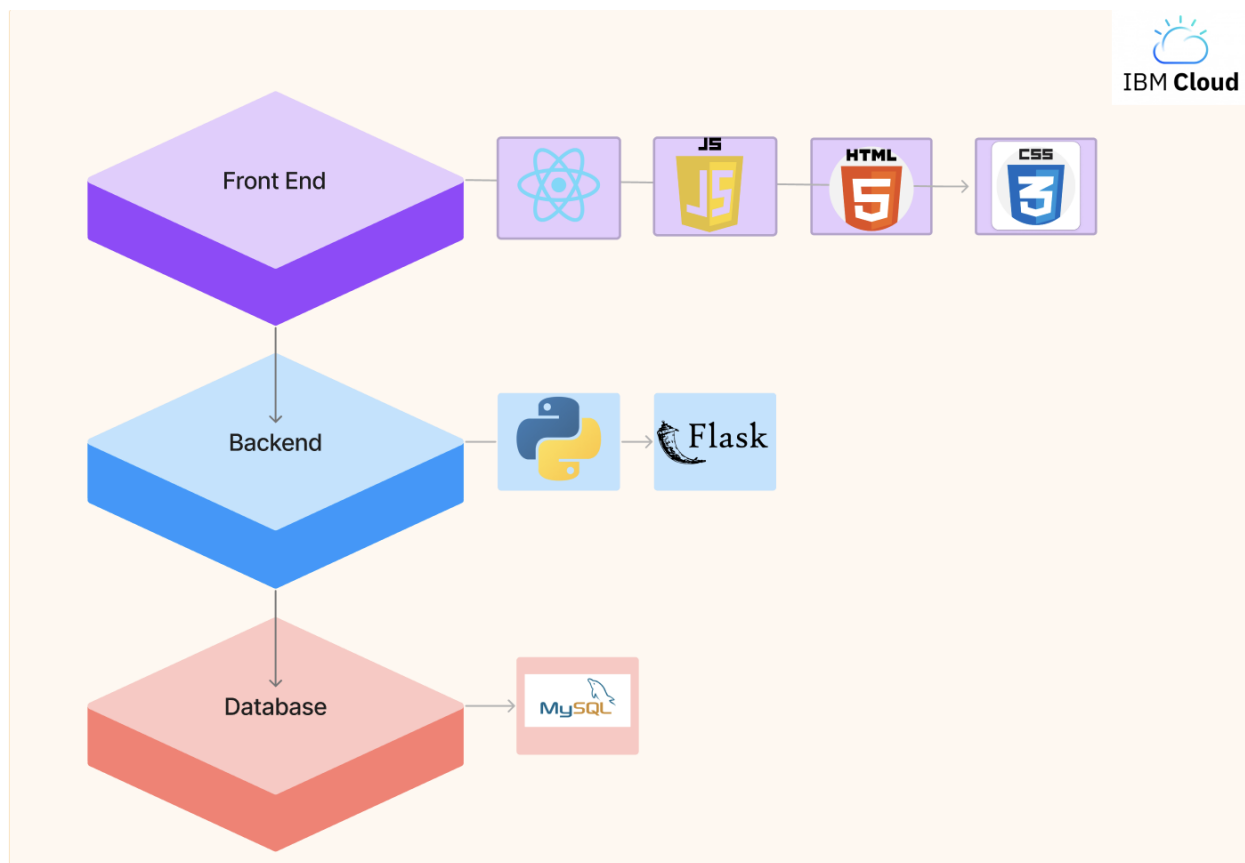
Lifespan of Data:

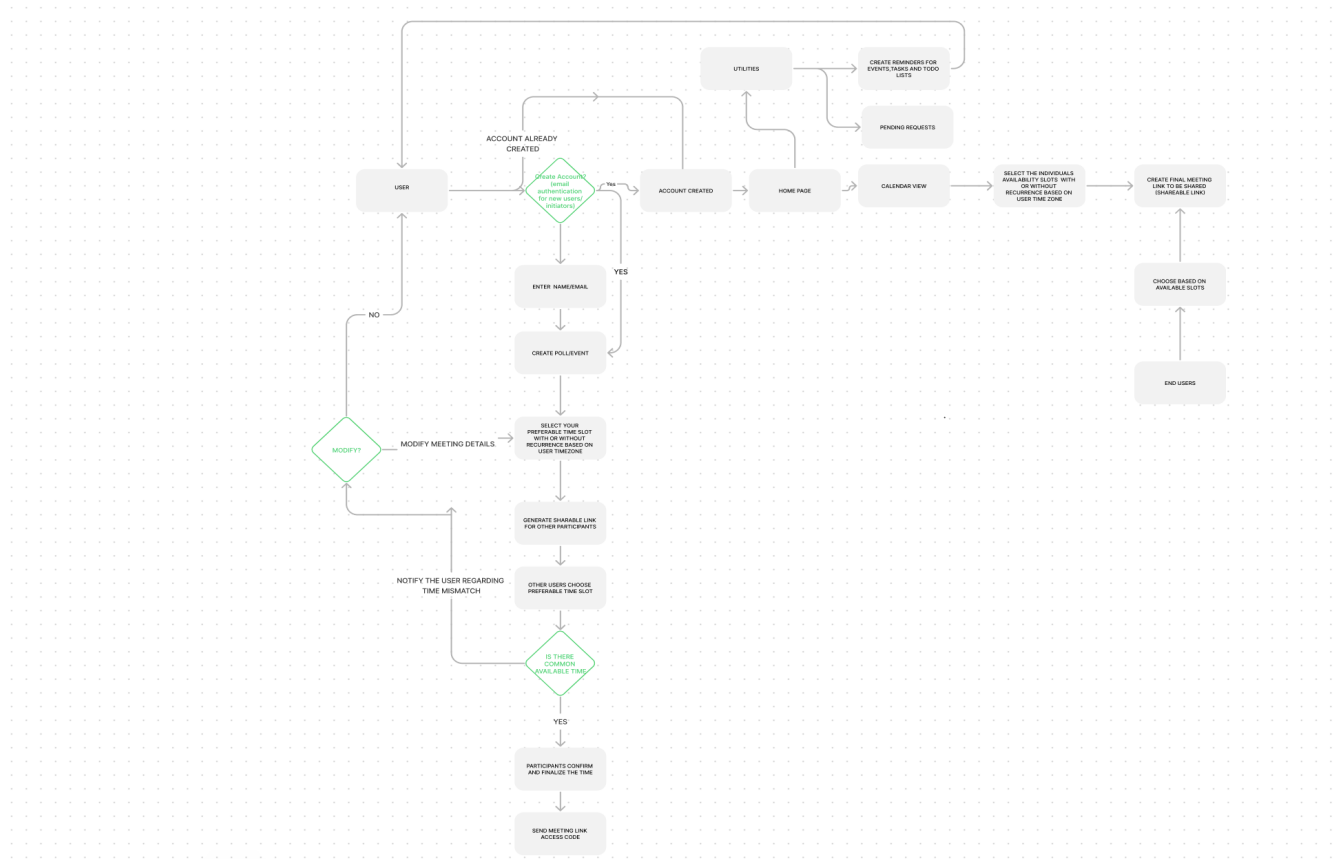
- Event data is stored for a required period of time after the event date to allow for review and analysis.
- Participant availability data is stored for a required period of time after the event date to allow for reference in future scheduling.
- After the specified time period, all data will undergo soft deletion

Data Removal:

- Deleted data is removed permanently from the system and cannot be recovered.
- The OES web application does not archive or summarize data, as it is designed to be a simple and lightweight scheduling tool.
- When data is removed from the system, it is deleted entirely to maintain user privacy and security.

System Architecture Diagram





Deployment Methodology

- When deploying a web application like OES, there are several steps involved.
- First, the code for the application must be built and tested to ensure that it is functioning correctly. This code will typically be hosted on IBM Cloud.
- Next, any necessary data for the application will need to be transferred to the production environment. This may include user data, such as profiles and preferences, as well as any other relevant data, such as scheduling information. This data may be migrated from the development environment to the production environment using a variety of tools and techniques, such as data backup and restore or data transfer protocols like FTP or SFTP.
- Once the application and data have been deployed to the production environment, it will typically undergo further testing and sanity checks to ensure that it is functioning correctly and can handle the expected load. This may involve stress testing, security testing, and other types of validation.
- As for the handover process, this will typically involve transferring ownership of the application and any associated intellectual property from the development team to the application administrator.

- With careful planning and execution, it is possible to ensure a smooth transition from development to production and to transfer ownership of the application and any associated data to the sponsor.