# Group 9: Sprint 9_ADR

## 1. Using Open-street Map Source with MapQuest vs Google Maps

In this sprint, the group had to select a navigating system platform. Google maps offers a Maps JavaScript API and SDK tools for web and mobile. MapQuest API offes JavaScript library (MapQuest.js) for interactive maps, geocoding, directions, and traffic. MapQuest.js is powered by the MapQuest APIs and integrates with Leaflet.js.

The decision to use MapQuest API over Google maps was because of accessibility. Google maps requires a developer to open a billing account which starts with a $200 (R4 112,30) credit. The credit decrements with use over time. Different pricing is listed https://cloud.google.com/maps-platform/pricing/.

It seemed a risk to use another platform on credit after our experience with choosing the Microsoft Azure student on a credit plan. MapQuest only requires creating an account and that gives you access to an API Key that gives you access to all JavaScript libraries mentioned above and 120 000 free transactions.

**Decision**

Use open street map source with MapQuest.

**Status**

Accepted

**Consequences**

Whatever the extent of our implementation with maps is or however vigorously we will be tested by the course co-ordinator and demi the platform will still be available.

## 2. Database schema changes

In this sprint, a couple of changes were made on the database schema. Firstly date, time and meetingStatus columns were added to the meetingRequests table. The date column stores the proposed meeting dates that users want to meet; the time column stores the prosed time; while the meetingStatus stores a boolena indicating if the user with a meeting was requested has chosen to accept or decline the invitation.

In the Resources table, the primary key Constraint was dropped because this constraint restricted different study groups from containing the same chat message.

A new table called userRating was created to store the overall user rating. The tables takes in the amount of stars (out of 5) a user has been rated and stores the average rating (also out of 5) in a column called ‘rating’. The rest of the other columns: oneStar, twoStar, threeStar, fourStar and fiveStar store the number of times a user was voted as oneStar or seconStar.

**Decision**

Alter database schema.

**Status**

Accepted

**Consequences**

These changes increase the expandability of the project. More features can now easily be added to the overall project.