



BITS Pilani

Software Architecture Assignment : 1

Submitted By:
Pavithra.S
2022MT93172



CLOUD REPORTING APPLICATION



PURPOSE OF THE SYSTEM



- The cloud reporting application is to centrally store all the testing data collected from a touch device and to automate the process of data collection by integrating directly with the touch device.
 - The touch device integrates with scanners that perform network tests and collects relevant data. This data is synced directly to the cloud data stores.
 - Our cloud platform will serve as a single serving platform for multiple type of end users having access to the platform to view, upload, configure and export test reports.
-

KEY REQUIREMENTS



1. Authentication:

- The user should be able to login into the application using an email/password combination.
- Based on the credentials, the user type is to be identified, and the corresponding admin or company view is to be displayed.

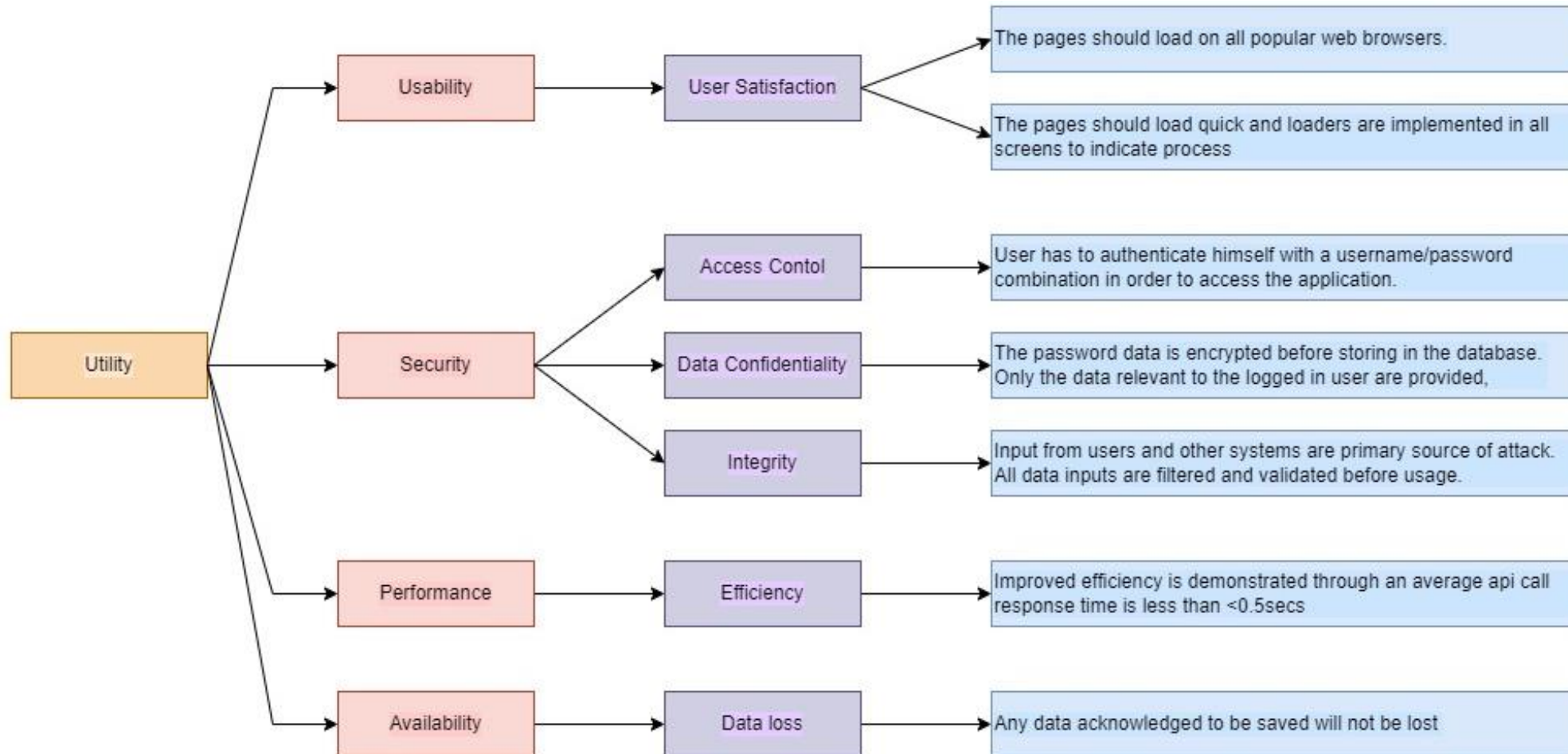
2. Admin Side:

- The super admin should be able to manage other admin users and their permissions.
- Create/Update companies and map subscriptions to them which will allow companies to manage building test data.
- Provide help to the companies through the support module.

3. Company Side:

- Add/Update company users and respective permissions.
 - Create or map building to the company.
 - Upload building test reports.
 - View and export the building test reports in pdf or word formats.
 - Create/Update test plan configurations.
-

UTILITY TREE



TACTICS USED TO ACHIEVE TOP 5 ASR'S



1. Usability

User satisfaction is achieved by,

- Making the system browser agnostic.
- Ensuring the page load time is less and a loader is displayed to inform the user about any background processes in the user interface.

2. Security

- Authenticate the user before the user access the system and check authorization before each request.
- The resources will run with minimum privileges to user accounts according to least privilege, based on a “need to know” approach.
- All inputs from users and other sources are filtered and validated to avoid security attacks.

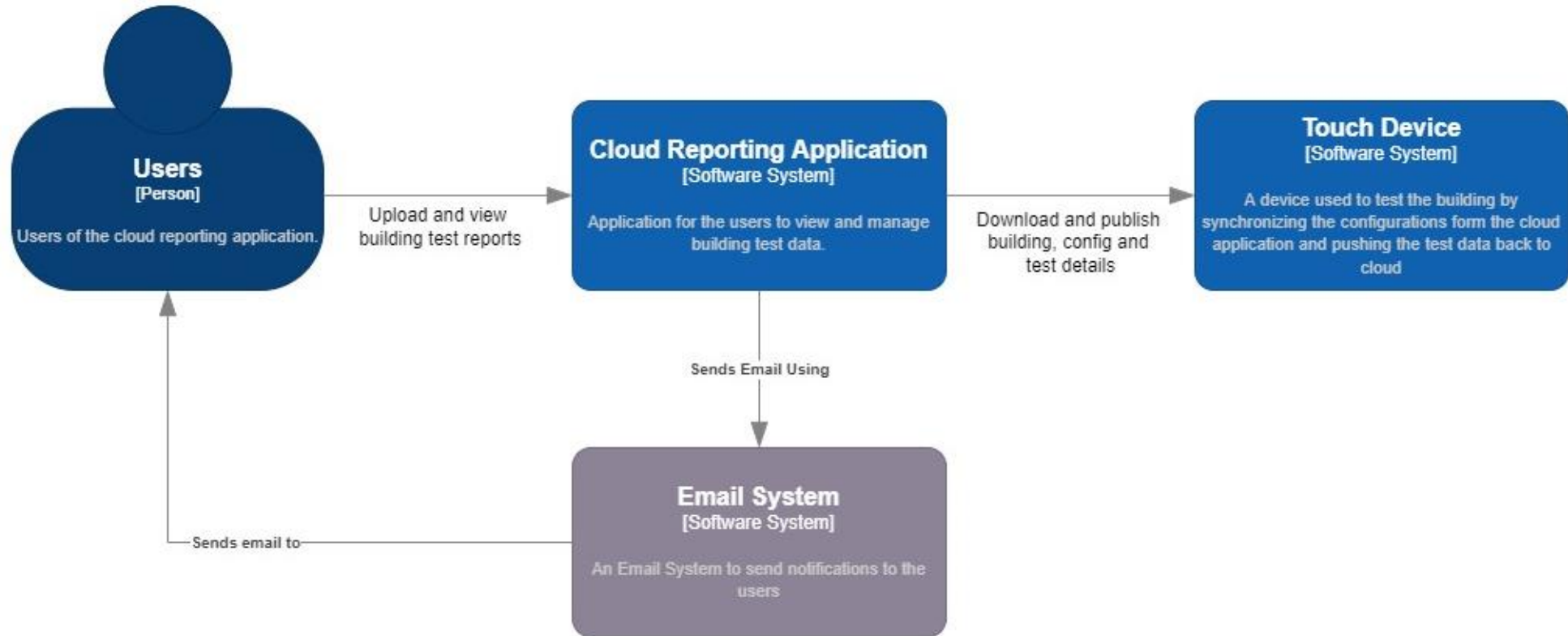
3. Performance

- The response time of the Api call is optimized as much as possible with an average of <0.5sec.

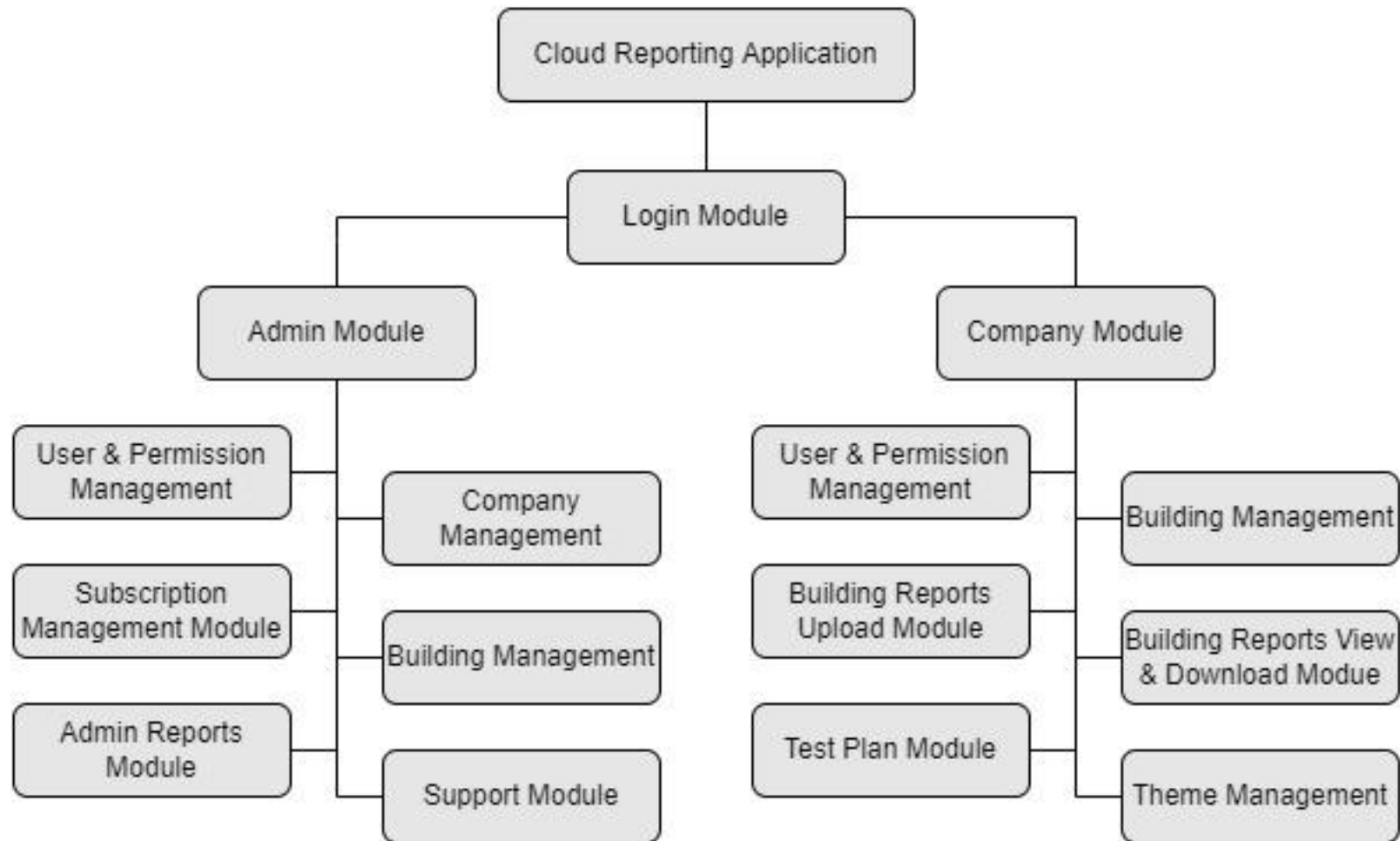
4. Modifiability

- The system is modularly decomposed to make sure that it can be modified easily.

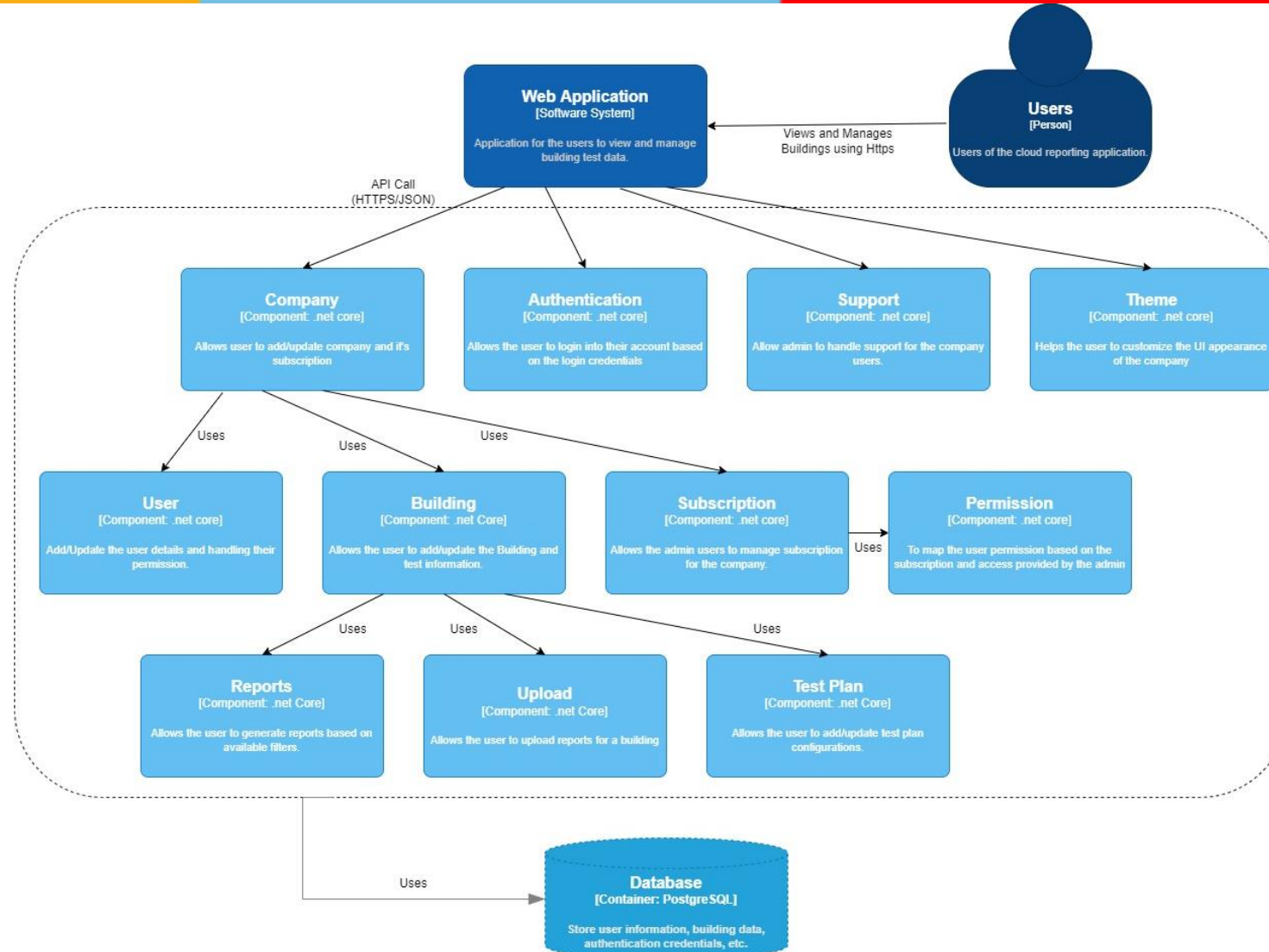
CONTEXT DIAGRAM



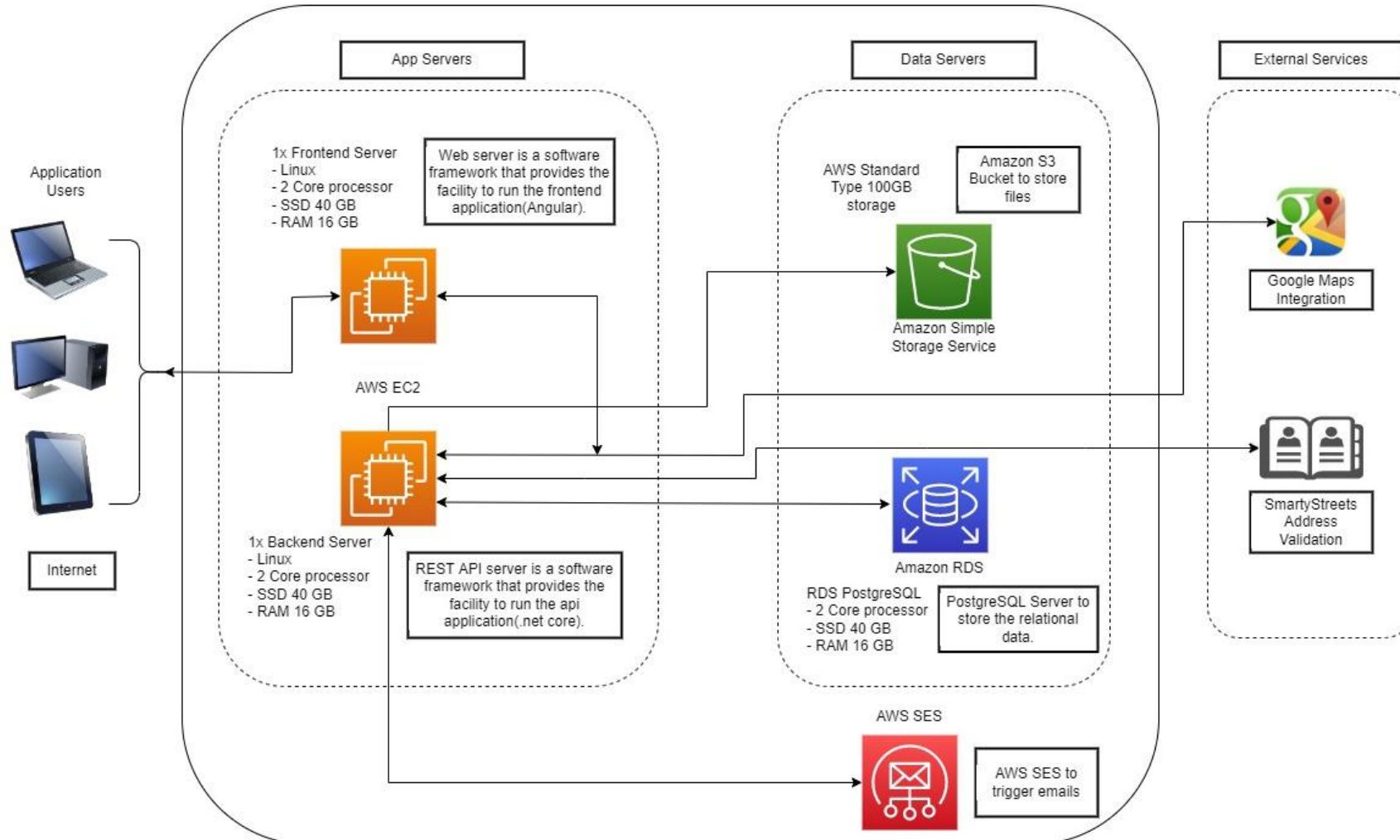
MODULE DECOMPOSITION DIAGRAM



COMPONENT & CONNECTION DIAGRAM



DEPLOYMENT DIAGRAM



HOW SYSTEM WORKS?



- The system follows a client - server architecture.
- Communication between the client and the server happens with REST services.
- The frontend angular application includes two views – admin view and the company view.
- The backend .net core application hosts a variety of Api's to fetch data from the database and save/update data to the database.
- The data is stored centrally. Files are stored using Amazon S3 bucket and all other data are stored in a PostgreSQL relational database.
- The data is synced between the cloud application and the touch device by having a common data store and using rest services to fetch/store data.
- The Amazon SES is used to send email notifications to the users.
- Since the application involves creating buildings for testing, google maps integration is done to get the pin location and SmartyStreets api is used for address validation.

KEY LEARNINGS



- For systems that involve many features, module decomposition should be planned accordingly.
- Planning the non-functional requirements such as scalability, availability, performance, security, modifiability, etc is equally important to planning the functional requirements.
- The system should be secure, and the data integrity and confidentiality should be maintained.
- It's important to consider the end user experience. The system should be easy to use and not compromise on the performance.
- The software architecture diagrams should be detailed enough to explain the architectural requirements.
- Network exposed services are frequent targets of attack and must be secured accordingly. Typically, such services are secured through appropriate authentication, authorization, and encryption controls.



THANK YOU