SATTELITE ALERT SYSTEM

By
Capt Zeeshan Ahmed
Capt Haseeb Ahmed
Submitted to the Faculty of Computer Science,
Military College of Signals, National University of Sciences and Technology,
Rawalpindi
In partial fulfillment for the requirements of a B.E Degree in
Computer Software Engineering
JULY 2017

TABLE OF CONTENS

Literature Review

Design & Development

Appendices

Analysis & Evaluation

Future Work

Bibliography

Abstract

A satellite is an object placed into an orbit around a planet. This object may be placed intentionally like artificial satellites or can exist in nature as is moon. Since 1957, thousands of artificial satellites have been launched from Earth and placed into orbit around the Earth. Artificial satellites originate from more than 40 countries and have used the satellite launching capabilities of ten nations.

Satellites are used for a large number of purposes. Common types include military and civilian Earth observation satellites, communications satellites, navigation satellites, weather satellites, and research satellites. The military uses include guiding the weapon systems and surveillance via various means and technologies i.e. cartography, oceanography, satellite imagery etc.

Satellite Alert System is developed in special context of the military uses of the artificial satellites, however the civil uses cannot be ruled out. The subject application is an active alert system based on tracking the orbits and footprints of hostile country's satellites which pass over various area of our country at different times.

Satellite Alert System is an SGP4 web application which can track satellites and display their respected orbits and show their passes and respective information. The project proposed is an idea to design a digital mechanism which will provide alert notifications to the users at the times of hovering and passage of hostile country's satellites over them at the present time.

Satellite alert system's significance over the existing alert mechanisms is that it is an efficient, time saving and easy way to get alerts and view the hovering satellites above the atmosphere. The aim is to develop a system that provides a platform for registering and recording available data of hostile Satellites and displaying their details when specified along with the alert of live satellite hovering over the atmosphere with its footprint on a live 3d map rather than reading lengthy and time consuming reports.

The idea is to incorporate the system in the Office Automation System's Portal of Pakistan Armed Forces for timely alerts and respective actions of the concerned departments to address the cause using a more advanced but easier approach and help understand the satellites motion and their surveillance instruments / technologies in a better way.

1. Introduction

In the perspective of spaceflight, a satellite is an artificial object which has been intentionally placed into an orbit around Earth. The world's first artificial satellite, the Sputnik 1, was launched by the Soviet Union in 1957. Since then, thousands of satellites have been launched into orbit around the Earth. Some satellites, notably space stations, have been launched in parts and assembled in orbit. Artificial satellites originate from more than 40 countries and have used the satellite launching capabilities of ten nations. About 6,600 satellites have been launched. The latest estimates are that 3,600 remain in orbit. of those, about 1,000 are operational; the rest have lived out their useful lives and are part of the space debris. Approximately 500 operational satellites are in low-Earth orbit, 50 are in medium-Earth orbit (at 20,000 km) and the rest are in geostationary orbit (at 36,000 km).

Satellites are used for a large number of purposes. Common types include military and civilian Earth observation satellites, communications satellites, navigation satellites, weather satellites, and research satellites. Space stations and human spacecraft in orbit are also satellites. Satellite orbits vary greatly, depending on the purpose of the satellite, and are classified in a number of ways. Well-known (overlapping) classes include low Earth orbit, polar orbit, and geostationary orbit.

1.1 Purpose

The main purpose of this project is to provide a solution to the users (Armed Forces / SPD etc) regarding the threats of the enemy satellites being faced by the country. This area has remained under shadow earlier and not much work has been done to let the users aware of this type of threat. So, the project brings the solution by providing a platform to the users from different domains including the army, navy, air force, Strategic Planning Division, Atomic Energy etc, which will bring awareness of the threat of satellites hovering above our country as well as will alert the users as per their requirements.

1.2 Document Conventions

1.2.1 SRS: Software Requirement Specification.

1.2.2 DB : Data Base.

1.2.3 SAS: Satellite Alert System

1.2.4 TLE: Two Line Elements

1.2.5 UI: User Interface

1.2.6 CZML: Cesium Language (the json schema the cesium viewer understands)

1.3 Intended Audience and Reading Suggestions

1.3.1 **Project Supervisor:** It will help to supervise the project and guide the team in a better way.

1.3.2 **Development Team:** It will help the developer to develop the product and to trace back the functional requirements.

1.3.3 **Testing Team:** It will help the testers to understand the constraints.

1.3.4 **Users:** The potential stakeholders of the system are the users who will be using the system.

1.3.5 **UG Project Evaluation Team:** Evaluation committee which will evaluate the progress of UG Projects.

1.4 Product Scope

SAS is a digital mechanism to provide users with a platform of tracking a required satellite and tracing its orbit and swath along with the provision of alert notifications of the satellites being tracked to the users at the times of their passage over the area of Pakistan. The notifications will be in the form of a pop up message which will be connected to a data base and at click will display the detailed information of the subject satellite.

1.5 Product Vision

Satellite alert system is an efficient, time saving and easy way to get alerts and track the satellites hovering above the atmosphere. The project is in context and based on evading the hostile country based satellites which monitor various important areas / installations and activities by various technologies i.e. cartography, oceanography, satellite imagery and various other information acquisition and surveillance means.

The notion of the project is to incorporate the system in the Office Automation System of Pakistan Army providing the staff officers an opportunity to get familiar with the concerned satellites orbits and swaths and their respective data along with being able to be provided with timely alerts.

LITERATURE REVIEW

2. Overall Description

2.1 Product Perspective

The project is a digital mechanism in context with the existing paper based information system of the concerned satellites which shall provide alert notifications to the user at the time of passage of the satellite being tracked. Moreover, the user will also be able to generate the swath / footprint of the satellite being tracked and also view its image, data and relevant descriptions.

2.2 Product Features

2.2.1 Track Satellites

SAS will display the movements of concerned satellites in the form of orbits and display its footprint when desired from a custom developed KML / SHAPE / CZML file along with their relevant details stored in the database.

2.2.2 Alert Notifications

SAS will generate alert notifications for the satellites being tracked and hovering above the atmosphere in the territory of Pakistan. These notifications will be displayed in the form of a pop up message and on click the system will display the detailed information about the selected or any satellite passing over the area.

2.2.3 Login/Access Right

The system will allow users to sign up and login to access the SAS at the startup page.

2.2.4 Database

This feature will allow Administrator to store and maintain records of the ID, name, image, instruments onboard and the technology used along with other relevant details of the satellites in the database.

2.3 User Classes and Characteristics

The software has two types of users, System administrator and regular user. Both types of users have different access level to the system and its data and can perform functions assigned to their respective roles.

2.3.1 System Administrator

The admin user can view all the users in the user records and can also delete them if necessary. He will have access to all the data records.

2.3.2 Regular User

The regular user will first sign up in the SAS and then log in the system in order to use is features with no access to the user records.

2.4 Operating Environment

SAS shall run on the computer system on an online web based system (Internet Browser) with;

2.4.1 Windows operating system

2.4.2 Pentium 4 or higher

SAS shall be managed with MySQL database.

2.5 Design and Implementation Constraints

Constraints of the product are given below:-

2.5.1 The availability of an internet connection.

2.5.2 The availability of the Keplerian elements in the form of TLE data and the accuracy of that data.

2.5.3 The implementation of TLE data in custom developed CZML file for each satellite.

2.6 User Documentation

Final release will be accompanied with a user guide to inform users how to use SAS. It will be available on the ABOUT page below the 'Documentation' heading to go through.

2.7 Assumptions and Dependencies

2.7.1 Basic assumption for development of SAS is that system should be available 24x7 since a user may wish to track a satellite at any time.

2.7.2 The server will be able to handle a large number of requests especially when initially entered, as there will be multiple track or query requests.

2.7.3 Overall performance of the product will depend on the hardware infrastructure and network speed.

2.7.4 User must be well conversant with UI for the better performance of the product.

2.7.5 Limitations of the accuracy of TLE data to be considered.

3. External Interface Requirements

3.1 User Interface

SMS consists of a web based interface that will be used by all the users. The learning curve for the interface should be gradual, so as to make the users of the application feel at ease while also enabling the Administration authority to interact with the system conveniently.

3.2 Hardware Interfaces

3.2.1 Computer System

3.2.1.1 System shall have keyboard input.

3.2.1.2 System shall have mouse input.

3.2.1.3 System shall have a monitor.

3.2.1.4 System shall have a working internet connection and the hardware requirements that come with it.

3.2.2 Database Server

To process requests and retrieve/store data.

3.3 Software Interfaces

3.3.1 SMS shall be able to deploy on active versions of the web browsers: Microsoft Internet Explorer, Mozilla Firefox, Google chrome.

3.3.2 Primary Operating System supported by SAS Interface will be WINDOWS.

3.3.3 SAS shall be deployed along with MySQL database.

3.4 Communications Interfaces

3.4.1 SAS shall be connected to the online logged in systems only.

3.4.2 Communication between the Web Interface and the server will be made using the traditional Client - Server Architecture.

4. System Features

4.1 Functional Requirements

4.1.1 The user shall be required to sign up into the SAS to use the application.

4.1.2 The user shall be required to Log in the SAS to access the application.

4.1.3 The system shall verify the login credentials from user table for logging in SAS.

4.1.4 The user shall be displayed the HOME page at the verification of the user credentials.

4.1.5 The system shall display an error message at the unsuccessful verification attempt of the user credentials.

4.1.6 The system will display the satellite movements on a world map in the form of paths or orbits in the map window at the HOME page.

4.1.7 The system will generate the swath (footprint) of the satellite when clicked upon the Footprint button.

4.1.8 The system shall display the name of the satellite being tracked on hovering of mouse cursor over it.

4.1.9 The system will generate the alert pop up notification upon the arrival of the hovering satellite over area of Pakistan.

4.1.10 The system will display the detailed information of the satellite including its image if available on click of the alert pop up notification.

4.1.11 The system will display the list of all the available satellites in the database on clicking the SATELLITES page.

4.1.12 The system will display the detailed information of the satellite including its image if available on click of the satellite name from the list at the SATELLITE page.

4.1.13 The system will display the information required by the user about the SAS project along with its documentation on clicking the ABOUT page.

4.1.14 The system will display the contact information to the user on the CONTACT page upon clicking the CONTACT page tab.

4.1.15 The system will display a search field along with a search button on clicking the SEARCH page.

4.1.16 The user shall be able to search the database through the search field by entering the satellite ID or NORAD ID or name or any relevant features.

4.1.17 The system shall display a 'NO MATCHES FOUND' message at the SEARCH page after a search is performed and the credentials are not matched or found in the system's database.

4.1.18 The system shall display a satellite / list of satellites at the SEARCH page after a search is performed and the credentials match with the data in the system's database.

4.1.19 An SAS administrator shall be able to add an account/member from the system's database.

4.1.20 An SAS administrator shall be able to delete an account/member from the system's database.

4.2 Non Functional Requirements

4.2.1 Performance/Time:

Processing Time: The processing time taken to display the orbit of a satellite on the map will be within 3 seconds.

Response time: This system will generate a quick response time as user requires an instant response similar to offline system as a query is made in the database.

4.2.2 Reliability:

The system will remain available at all times and shall be tested against the TLE data.

4.2.3 Security:

The system shall not allow unauthorized access to the system and its data.

4.2.4 Accessibility:

The system shall be accessible at all times to the user online.

4.2.5 Platform independence:

The user shall be able to use the system from any browsing platform.

4.2.6 **Maintainability:**

The system shall be maintainable in order to accommodate any minor changes in the functionalities or fixing the reported bugs

DESIGN & DEVELOPMENT

1. Overview

The System architecture description section is the main focus of this chapter. It provides an overview of the system's major components and architecture, as well as specifications on the interaction between the system and the user.

This document is organized in such a way that the detailed architecture of the system is provided initially. The design is further elaborated using diagrammatical representation of system components, classes, states, sequence of events, flow of events and their relationships. Furthermore user interface constraints have been discussed along with issues like product reuse, design decision and tradeoffs, pseudo code for components and appendices.

The Section is divided into following sub-sections and is already listed in the table of contents and figures list. However, here is a brief description of all the sub-sections.

1.1 System Architecture Definitions

In this section, the overall architecture of the system is discussed, including the introduction of various components and subsystems. It is mainly supported by system Architecture diagram which shows an insider's perspective of the system by describing the high level software components that perform the major functions to make the system operational.

1.2 Structure and relationships

This section ponders upon the interrelationships and dependencies among various components. It is mainly described by a diagram which is further augmented by explanatory text. UML Class diagram also helps us understanding the system structure.

1.3 UML Class diagram

UML Class diagram further manifests the description of low level components of the software that include data storage and state details, thus making the system adequately comprehensible. (page 19)

1.4 User Interface Issues

This section presents the main principles of the product's user interface. Not touching about the technical details, the section is described by an overall diagram which is also augmented by explanatory text. Moreover, UML Activity diagrams, UML Sequence diagrams, and UI Design diagrams also elaborate the User Interface issues in a more intelligible manner.

1.5 UML Activity diagrams

UML Activity Diagrams follow a workflow-based approach to describe the overall functioning of the system. They are a very good means to see how various steps are involved in major tasks inside a system using a flow chart pattern without getting into the technical details. (page21)

1.6 UML Sequence diagrams

UML Sequence diagrams show how different objects are involved in the completion of a functionality of the system. They have a unique format that allows the reader to see how many objects are used vis-à-vis their duration; for the completion of a system requirement.(page23)

1.7 UI Design

Some screenshots of graphical user interfaces are shown in this section that prototype the way a user shall be interacting with the system.

1.8 Detailed description of components

This section contains detailed description of all the major components of the system in a structured pattern (table), comprising of 10 x rows. The pattern (table) maintains symmetry in the document structure; and therefore it is followed for each of the components. Each part/row of the table is identified by a label, explaining the purpose of each point. The description of each point vis-à-vis the component being discussed, ponders upon the detailed account of it in the system.

1.9 Reusability and relationships to other products

This section focuses upon the Reusability aspects of the various components of the system. Since the project in hand is all new and doesn't carry out any enhancement work in the already existing system, so Reusability is just a recommended strategy to be employed while organizing various system components.

2. SYSTEM ARCHITECTURE DESCRIPTION

2.1 System Architecture

Client-Server Architecture will be used to implement SAS. From a high level perspective, a service-based solution can be seen as being composed of multiple services, each

communicating with the others by passing messages. Conceptually, the services can be seen as components of the overall solution. However, internally, each service is made up of software components, just like any other application, and these components can be logically segregated into 3 modules at the “server-end” where the system will be deployed.

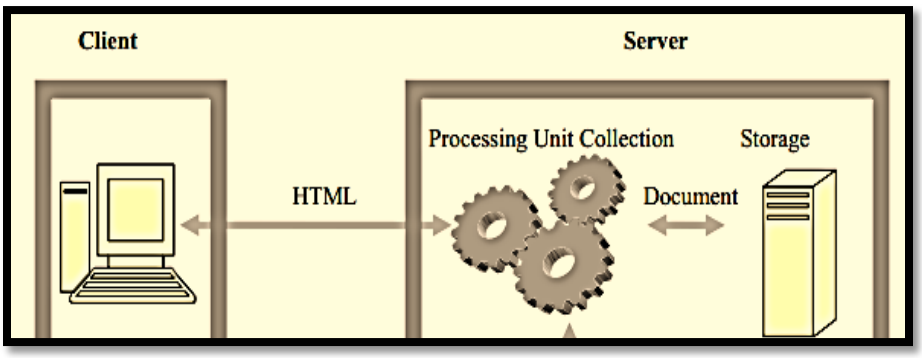


Figure 1 - System Architecture

2.2 Overview of the modules

The system will be architected mainly in 3 fundamental modules “User Interface”, “Core System”, and “Database”. The “Core System”, and “Database” will come under the “Main Module”. The “DB” or “Database” will have further two sub modules (Satellite DB and User DB) as shown in the following abstract diagram:

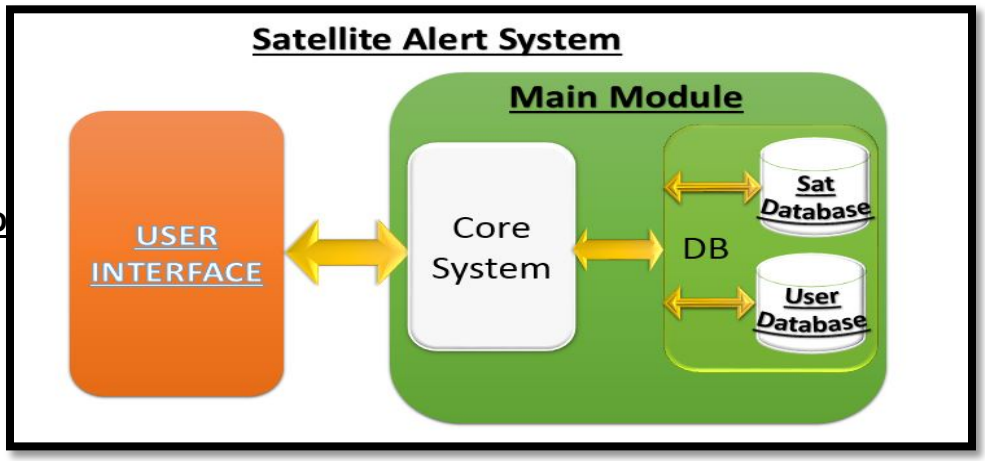


Figure 2 - Overview of the Modules

2.3 D

The “User Interface” will be used to pass on the information to “Core System” inside the “Main module” which will determine the service required. It will interact with the “DB” in order to exchange the information with the

respective database. The DB module will fetch the requisite information and will pass this information to the “Core System” for further processing.

2.4 Component Details

The details of the components are discussed below.

2.4.1 User Interface

User Interface caters for the visual needs of the application, wherein the Human-Computer-Interaction aspects are considered to enable the user to communicate with the system profoundly. It is connected with the main module through the “Core System” to pass on the user inputs to it as well as displays the output received from the “Core System” to the user respectively. It consists of following components:

2.4.1.1 Login

This component precedes all the requests by the user and hence, all the activities of the system by asking for the login info and validating it via the *Authentication* component of the Core System. Failing which will render the system to the halt state and will it re-ask for login credentials.

2.4.1.2 Home

This component acts as the main display page or the landing page with requisite login. This page shows the list of satellites into the system, the *Map Window* and *Satellite Info* (window). It interacts with the *Core System* module via *Main Window* component.

2.4.1.3 Admin

Component deals with the administration of the system by managing the users as well as the satellite data entry / deletion via standard entry form.

2.5 Core System

The Core System comprises all the services provided by the system including the Base-map, Sat TLE, Alert, Sat Management, User Management, Authentication sub-modules.

2.6 Map Window

This sub-module gets the data from base map, Sat Info and Alert components and applies SGP4 Algorithm for the satellite motion on the TLE data fetched via *Sat Info* component. The output is provided to the *Home* component of the UI. Base map Component provides a "Cesium" based base map. Which offers a 3D Globe of the world for mapping purposes as shown:-

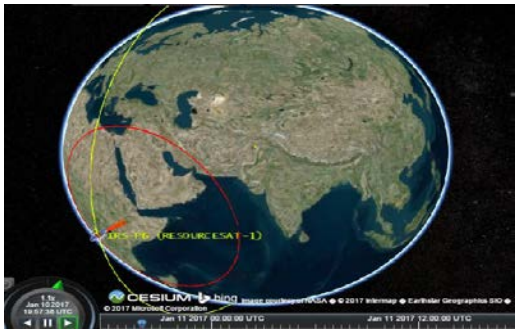


Figure 3 – Cesium Map

2.7 Sat TLE

This component accesses the TLE data of respective satellites from the *Sat DB* and feeds the *Main Window sub-module.*

2.8 Alert

This component is used to send alert messages.

2.9 User Management

This component is used by "administrator" to manage User accounts. It acts as mediator between the *User DB* and the *Admin*.

2.10 Sat Management

This component is used by "administrator" to manage satellite data. It acts as mediator between the *Sat DB* and the *Admin*.

2.11 Authentication

This component is used to authenticate the users. It facilitates the system by validating the existence of a user account in the *User DB.*

2.12 DB System

This is the database of the whole software system. All the “user” as well as “satellite” related data is maintained herein under two different databases named as Sat DB and User DB. It has been designed by keeping data integrity and confidentiality principles in mind. It serves with request and response mode.

Structure and relationships

Focusing upon the internal structure of the system, this section ponders upon the interrelationships and dependencies among various components.

3. Overall Structure of the system (Chart)

The diagram shows the main components of the system along with their interactions with each other. It mainly describes the system structure which is further augmented by the explanatory text as follows:

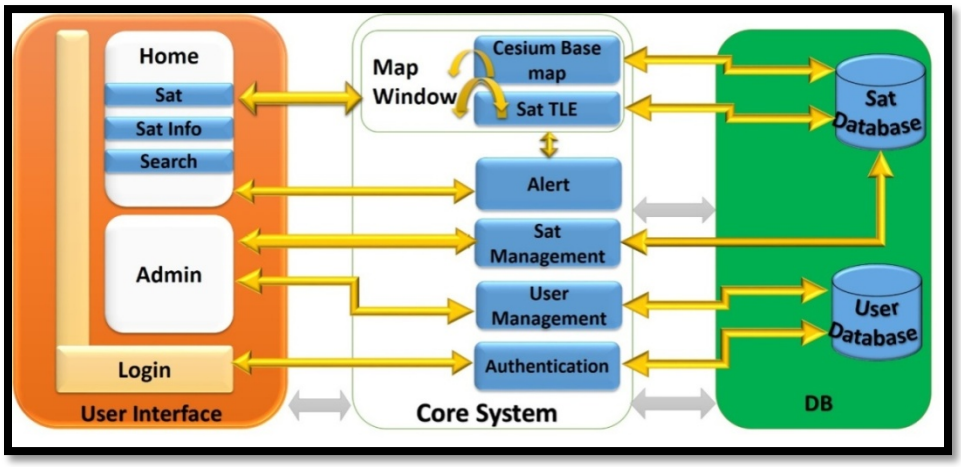


Figure 4 - Overall Structure of the System

3.1 Relationships of Components

The operational relationships amongst different components have been shown with the following state-machine diagram:

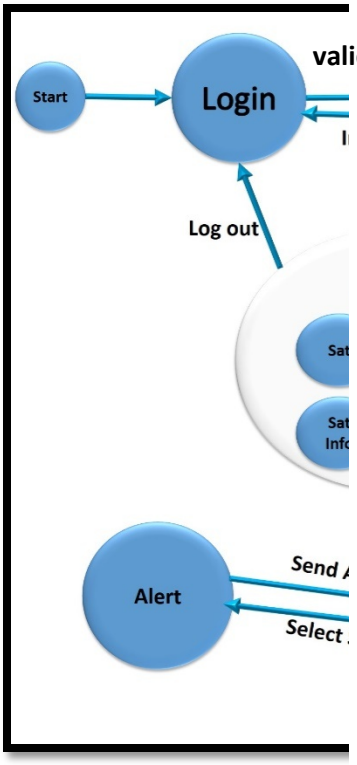


Figure 5- Operations of the product components.

4. User Interface design

4.1 Use Case Diagram

Administrator

User

Figure 6 - Use Case Diagram

4.2 Class Diagram

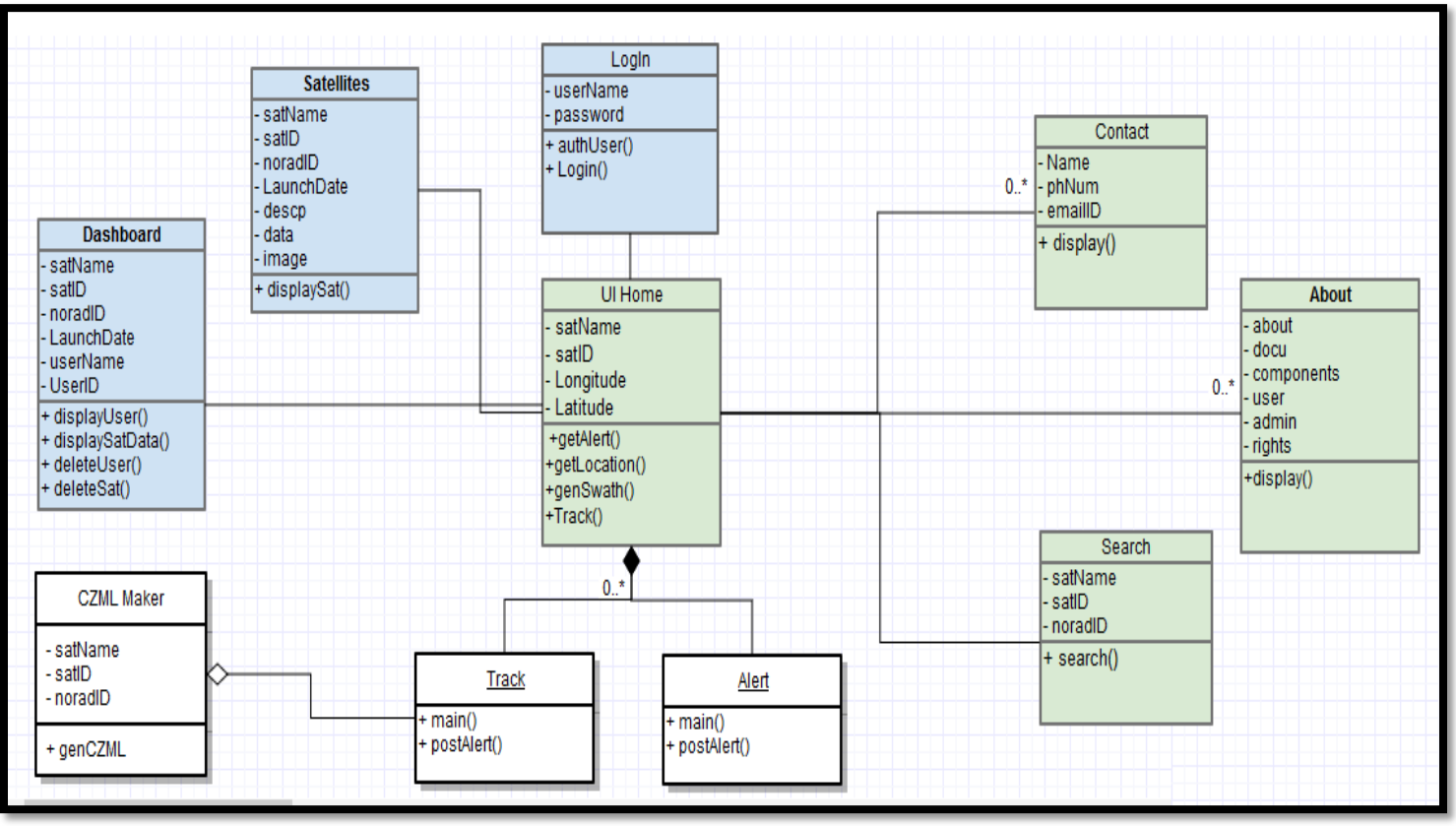


Figure 7 - Class Diagram with Description

4.2.1 Description of the diagram

All the classes shown in the class diagram are further described in the sub sections.

4.2.1.1 UI Home <<Interface>>

Contains all the functions which SAS has to offer. It calls all the functions from respective classes as and when invoked by the user.

4.2.1.2 Login

This is the main page to access the SAS application. The data operations performed are user authentication and consequent login or error message and sign up as a new user.

4.2.1.3 Satellites

This class contains the information about all the data about the satellites available in the database.

4.2.1.4 Search

It contains the search functions required for searching the concerned data about the satellites by the user.

4.2.1.5 About

This class has information about the documentation and working of the web application which the user may want to know to use the SAS web application.

4.2.1.6 Track

This class contains data and functions for tracking the marked satellite by the user and to display it on the cesium viewer.

4.2.1.7 Alert

This class matches the swath of the passing satellite with the location of the user and delivers him a pop up alert notification.

4.2.1.8 Dashboard

This class the information about all the users and satellite data saved in the respective Databases and displays it to the administrator.

4.3 UML Activity diagrams

This section shows the activities that a user need to perform to accomplish a task.

4.3.1 Login

Description: This scenario describes the flow of activities necessary for the user to either sign in or sign up in the system.

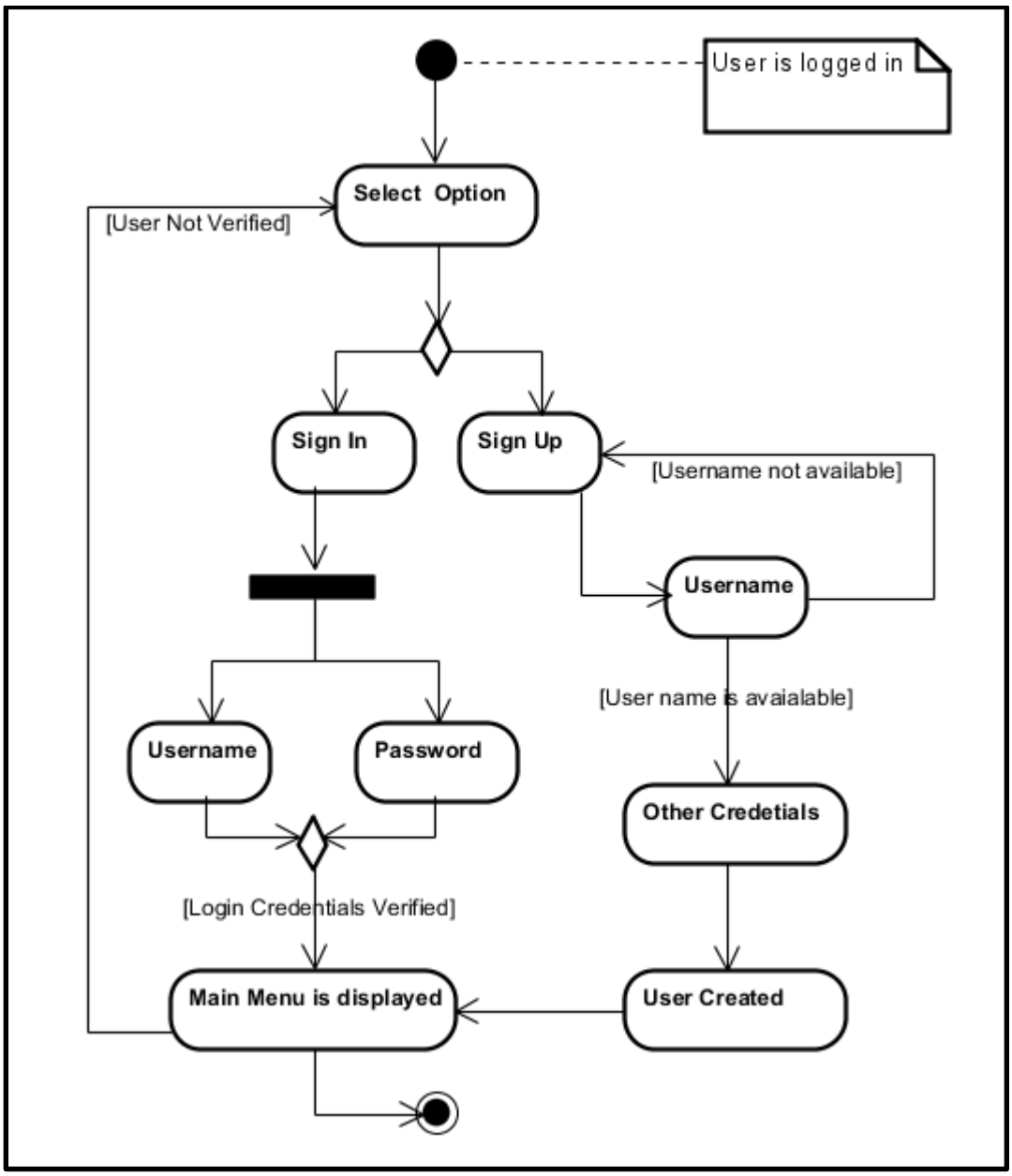


Figure 8 - Login Activity

4.3.2 Track Satellite

Description: This scenario describes the flow of activities necessary for tracking the satellites and getting alerts.

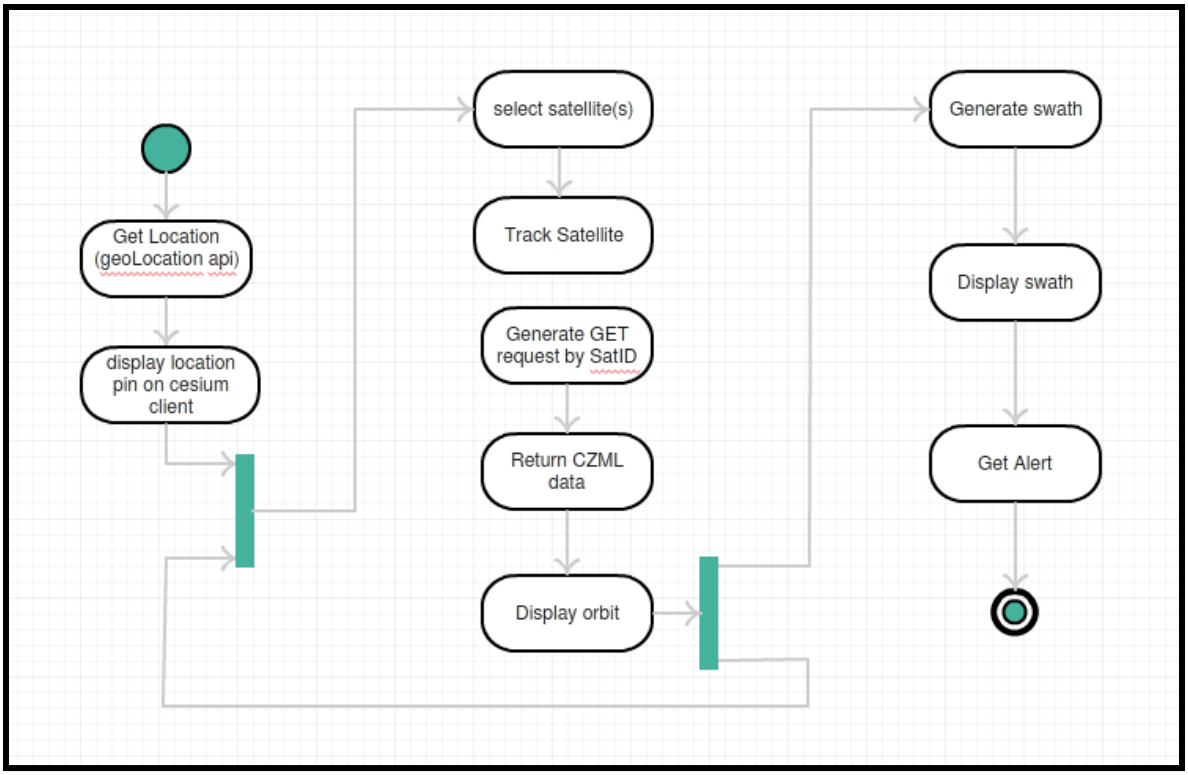
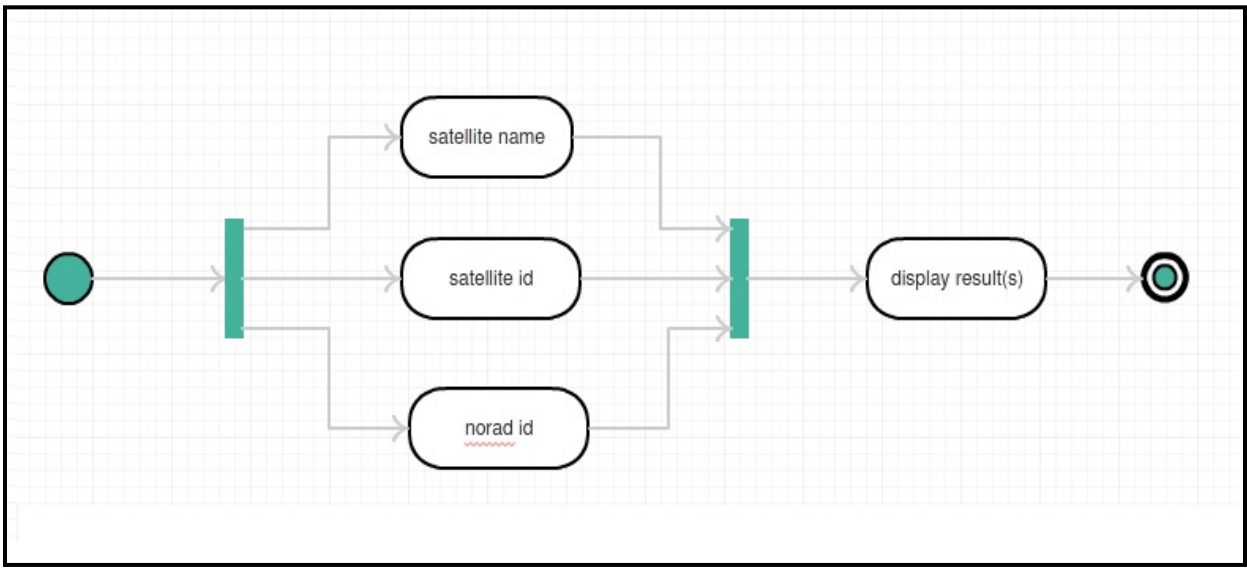


Figure 9 - Edit Profile Activity

4.3.3 Search

Description: This scenario describes the flow of activities required to search the satellites database.

Figure 10 - Search Activity



4.3.4 Satellites

Description: This scenario describes the flow of activities at the Satellites page.

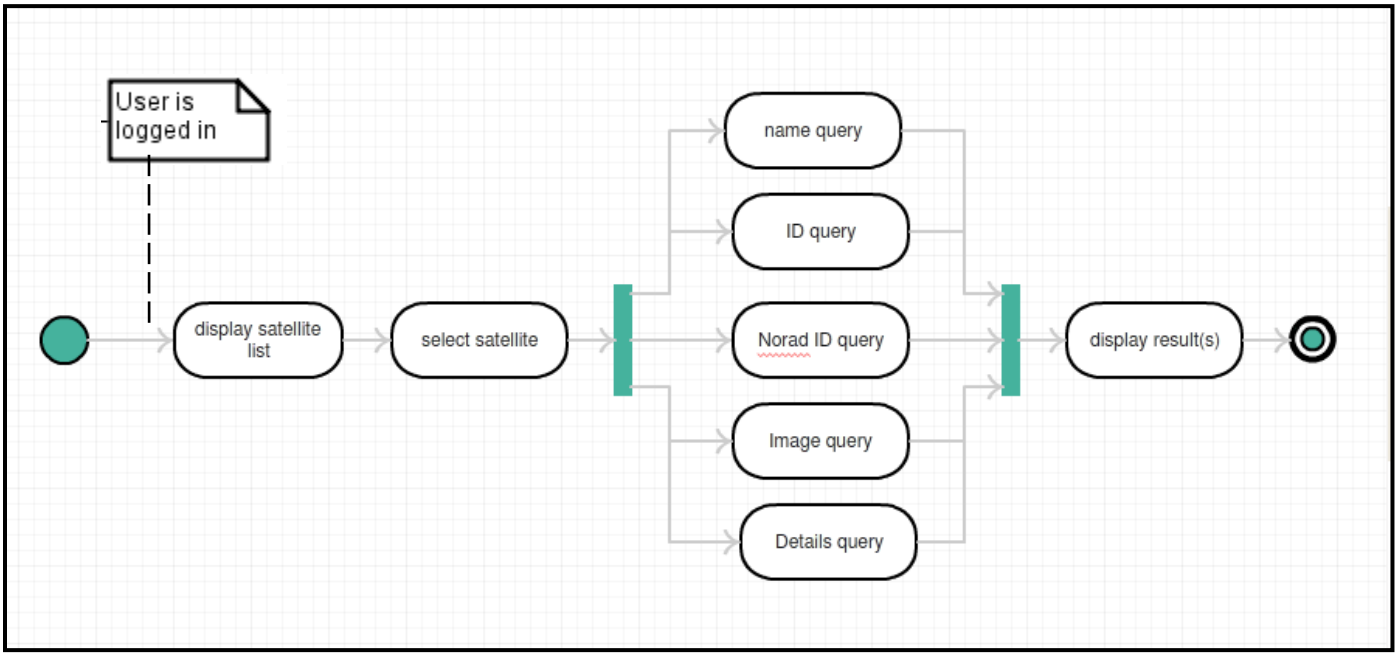


Figure 11 - Satellites Activity

4.4 UML Sequence diagrams

Different Scenarios and their corresponding events are discussed in this section with the help of sequence diagrams.

4.4.1 Login

Description: This scenario describes the sequence of events that take place when either the users logs in to the system or signs up in the system. The alternative prospects of the events have also been catered for in case the user tries to log in without valid credentials, or if the user tries to sign up with an existing name.

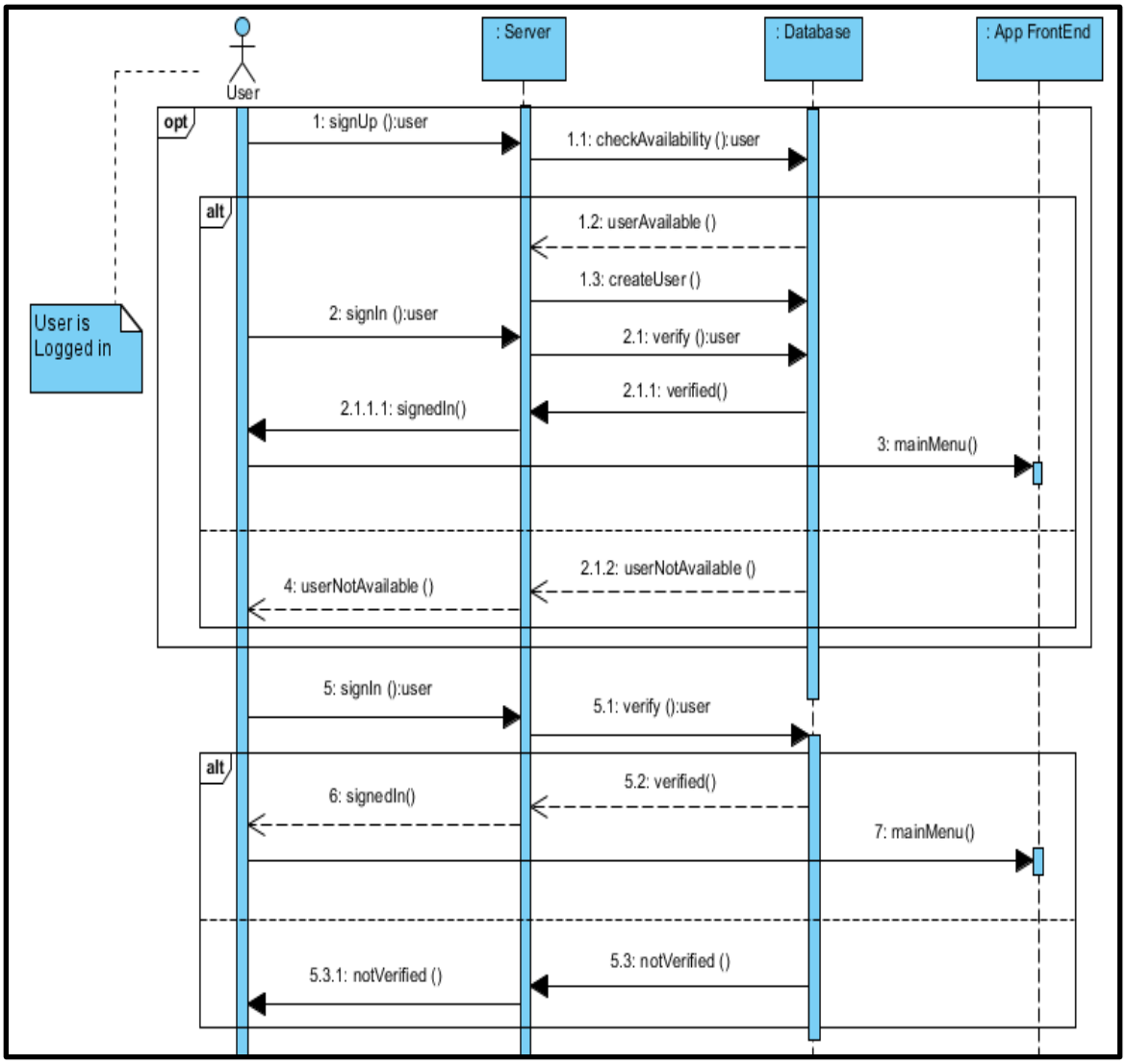


Figure 12- Login Sequence

4.4.2

View-Edit Profile

Description: This scenario describes the sequence of events that take place when a logged in user intends to edit his profile, including name, contact and service details; etc.

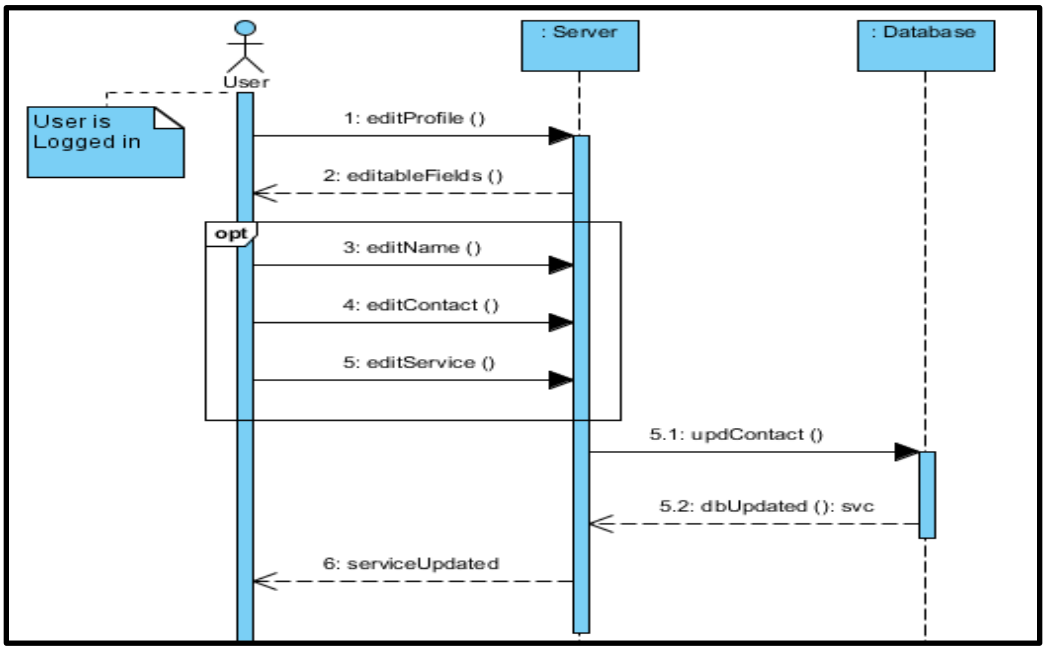


Figure 13 - View-Edit Profile Sequence

4.4.3 Track Satellite

Description: This scenario describes the sequence of events that take place when the user intends to track a satellite and get alert notifications.

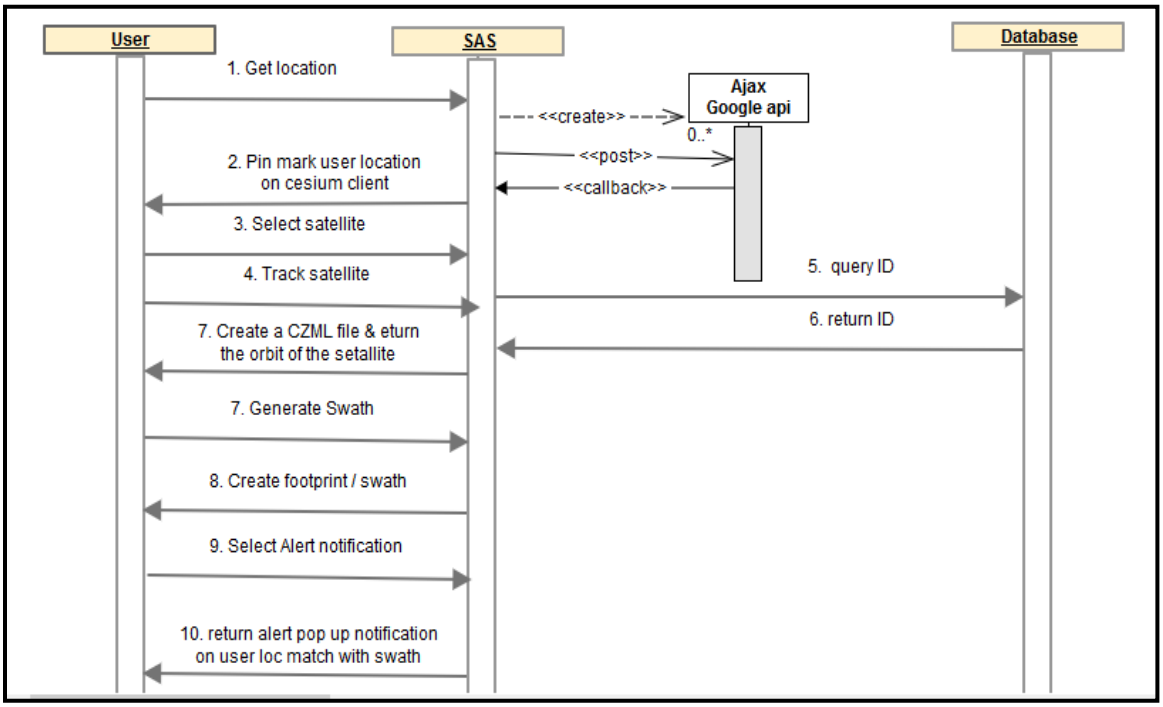


Figure 14 – Track Satellite Sequence

4.4.4 Search

Description: This scenario describes the sequence of events that take place when the server gets a search request by the user and its subsequent results.

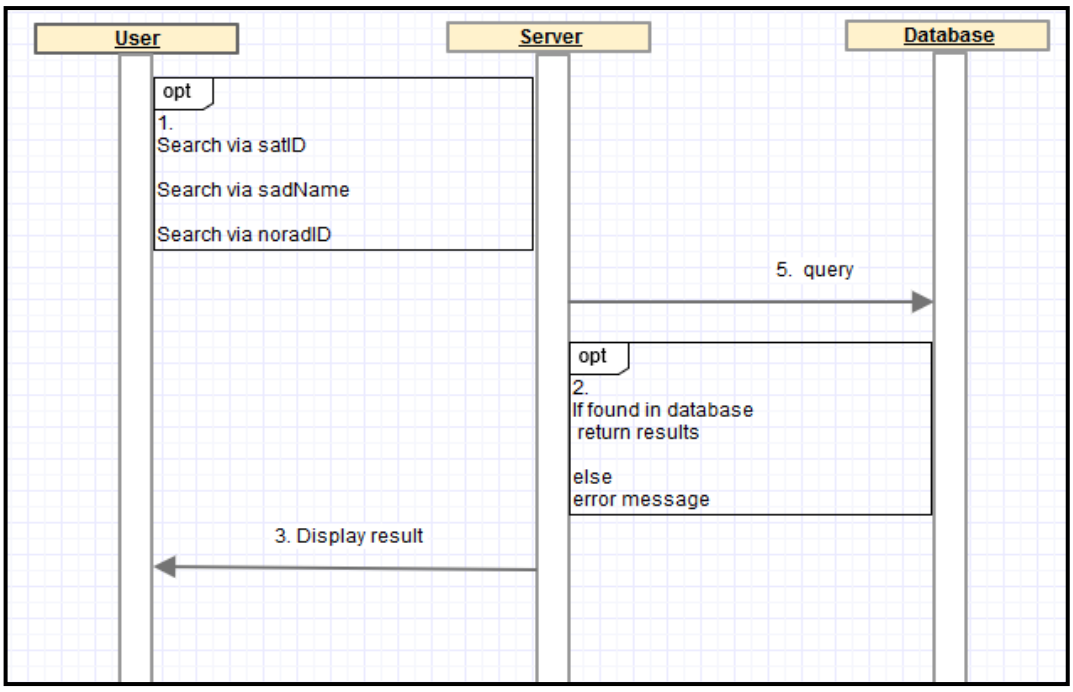


Figure 15 - Search Sequence

5. UI Design

SAS is a web application that shall consist of a web based interface that will be used by all the users. The first screen will be the login page after which the system will display the home page, a map centered page with title at top and page buttons stacked at the left to traverse within SAS and the satellite names.

5.1 UI home page

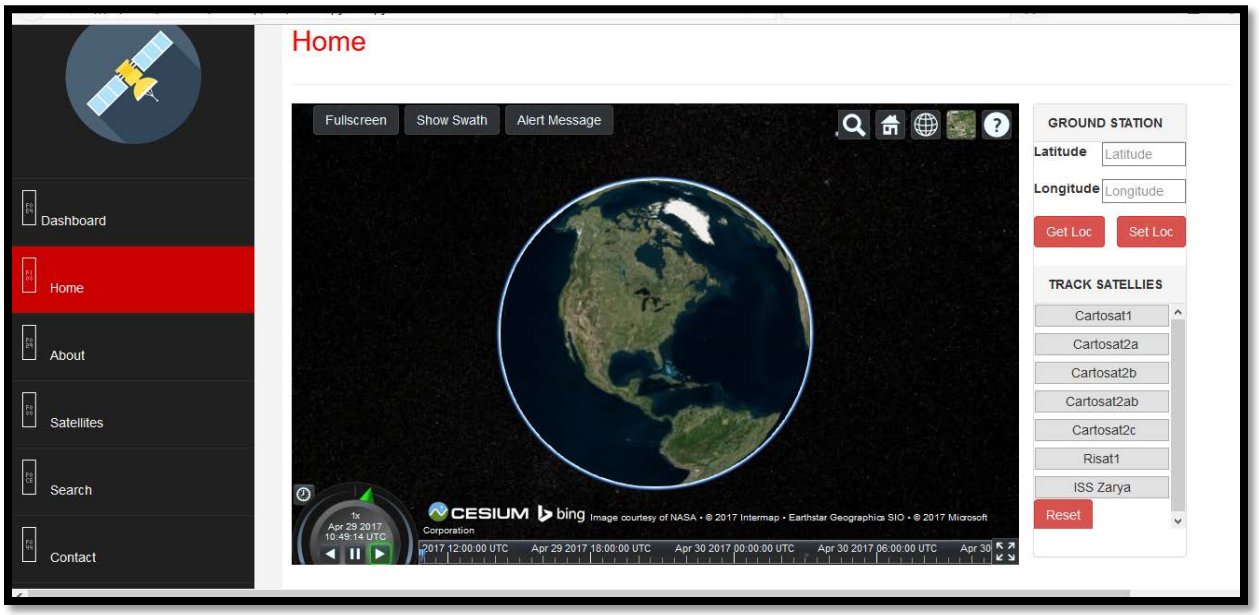


Figure 16 – UI Home - UI Design

5.2 Satellites page

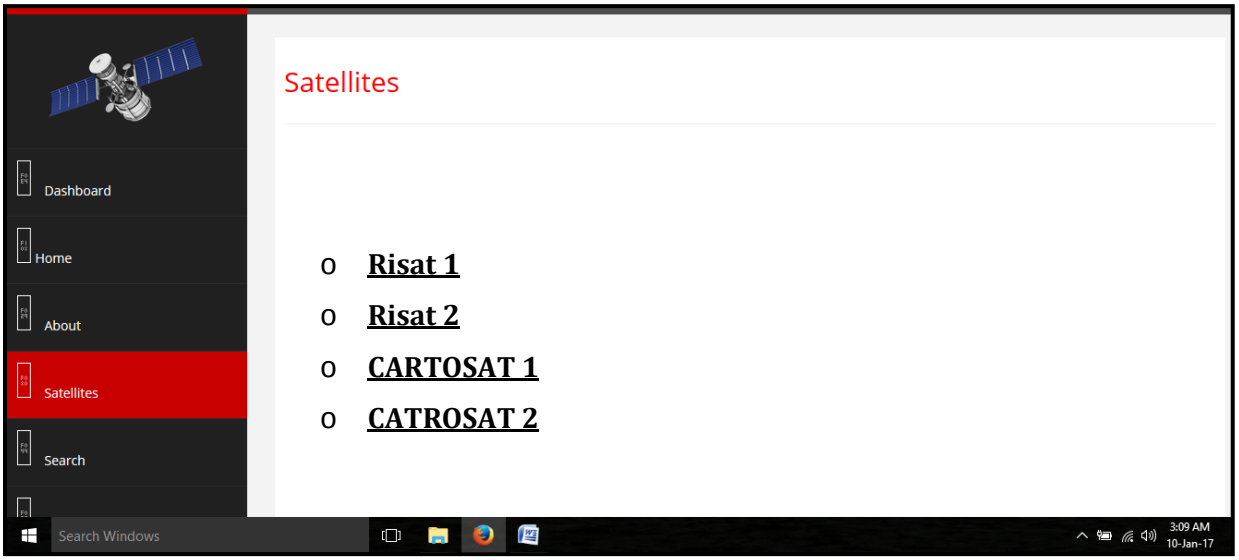


Figure 17 - Satellites - UI Design

5.3 Search

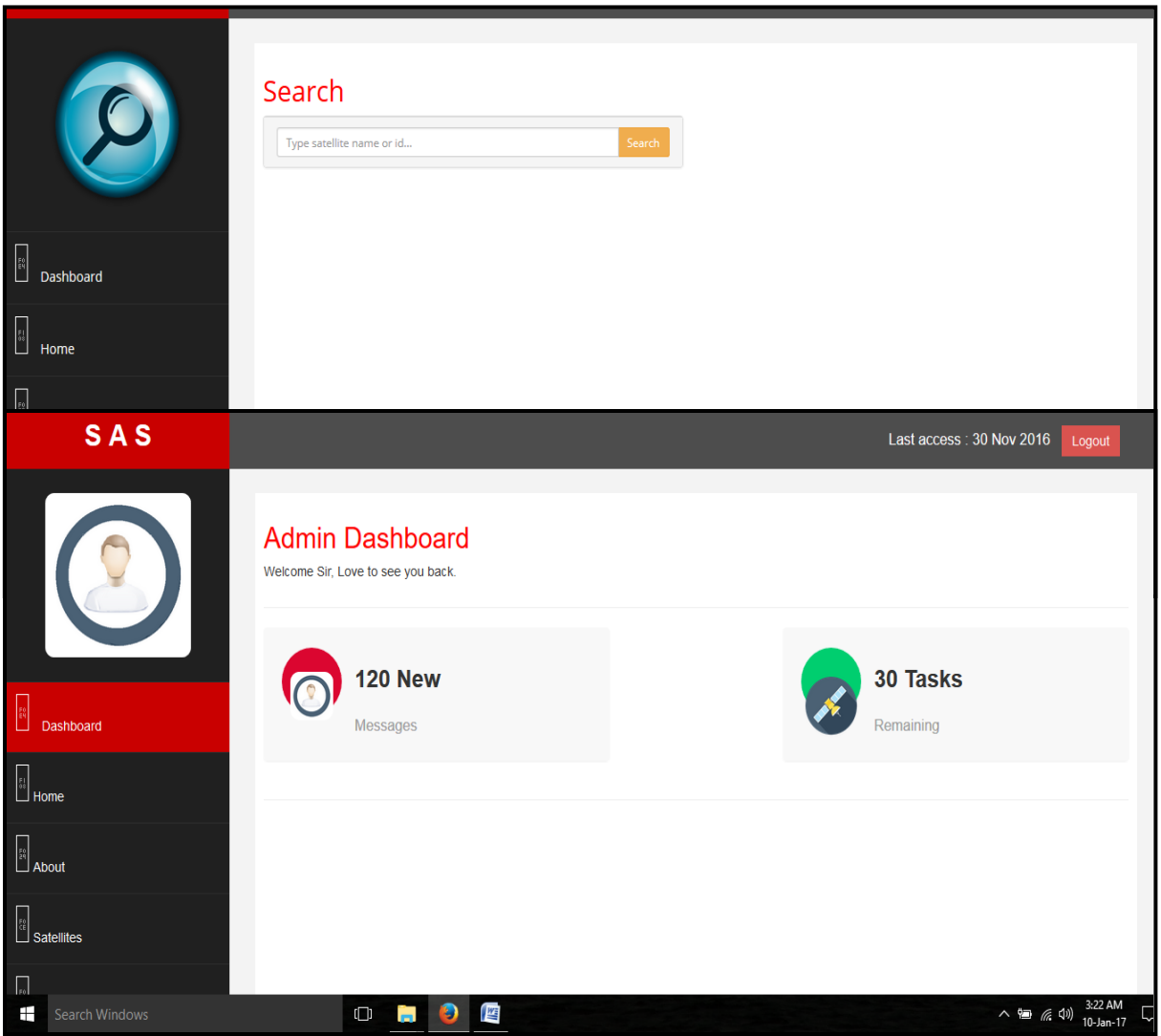


Figure 18 - Search - UI Design

5.4 Dashboard

Figure 19 - Dashboard - UI Design

6. DETAILED DESCRIPTION OF COMPONENTS

6.1 UI Home

Identification	UI Home
Type	HTML
Purpose	User interface to access different features of the SAS. Separate menu instances for each menu screen.
Function	Displays a page with buttons representing different menu choices to traverse and a Cesium Viewer centered page with Satellites list on the right to track and display the orbits of satellites and ALERT button below to get the pop up notifications.
Subordinates	Html hyperlinks / buttons, one for each page of the SAS application and other tasks.
Dependencies	Traverses to the page of hyperlinked button. Invokes the functionality of the pressed button.
Interfaces	Pages (hyperlinks) Satellite names (hyperlinks/radio buttons) Cesium viewer client Alert button (html button) Track button (html button) Swath button (html button)
Resources	Web browser
Processing	Pages (hyperlinks) to traverse to that page Track button (html) to display the orbit of satellite trackSat()

	Alert button (html) to get the alert notification of the satellite being tracked. alertNotification() Swath button (html) to display the footprint of the satellite being tracked. Swath()
Data	CZML file

6.2 Satellites page

Identification	Satellites
Type	HTML
Purpose	User interface to access different pages of the SAS and enable user to be provided with essential data on satellite details.
Function	Displays a page with buttons representing different menu choices to traverse and a list of satellites in the center of the page.
Subordinates	Html hyperlinks
Dependencies	Traverses to the page of hyperlinked button. Invokes the functionality of the pressed button.
Interfaces	Pages buttons (hyperlinks) Satellite names (hyperlinks)
Resources	Web browser, database
Processing	Pages (hyperlinks) to traverse to that page Satellite names (hyperlinks) to display the detailed information of the satellite.
Data	Name, id, detailed info(String)

	Image(jpeg, gif, png)

6.3 **Search**

Identification	Search
Type	HTML
Purpose	User interface to access different pages of the SAS and allows the user to search the satellites database.
Function	Displays a page with buttons representing different menu choices to traverse and search satellites database
Subordinates	A search bar with search button will search the database tables.
Dependencies	All search queries are dependent on this class for their display to the screen.
Interfaces	Search bar (html text field) Search button (html button) search()
Resources	Web Browser , database
Processing	Search bar (html text field) to write the search query value Search button (html button) to invoke the search function to query the database.
Data	satName, satID, noradID (string)

6.4 **Dashboard**

Identification	Dashboard
Type	HTML
Purpose	User interface to access different pages of the SAS and image hyperlinks

	for the admin to access the system databases.
Function	Displays a page with buttons representing different menu choices to traverse and allows the administrator to manage the system databases : 1. User database 2. Satellites database
Subordinates	Data tables
Dependencies	All User / data management functions depend on it.
Interfaces	Images (hyperlinks)
Resources	Database
Processing	On pressing the image of both user database and satellites database the administrator will be displayed with the table of that category.
Data	Table (user database) Table (satellites database)

6.5 **Login**

Identification	LogIn
Type	Class
Purpose	To manage the registration of Users, their User Credential Database and validation of User Credentials in case of Login activity. All of this functionality is hidden from the user.
Function	Represents options to the user, i.e. signup() login() verify()
Subordinates	User credentials, used to authenticate the user for login. Services data to help user finding the service records.

Dependencies	All login activity depend on it.
Interfaces	login(user) verify(user) signup(user)
Resources	Database
Processing	signup () signs up the user by valid credentials for onward login login ()authentication of the valid user for allocating access rights verify(user) verify user credentials from the user database
Data	Database table for maintaining the user record.

7. REUSABILITY & RELATIONSHIPS TO OTHER PRODUCTS

The SAS web SGP4 application is a new product and not any enhancement being done in an already existing product. Therefore no explicit component is being reused except the comprehensive base map of the world which has to be availed for use. However, the strategic aspects of future reusability of its components are supposed to be considered. Therefore, right from the architecture, the system is being designed in a modular fashion with a view to have more cohesion and less coupling among the sibling modules.

8. PSEUDO CODE OF COMPONENTS

8.1 Login

While User not signed in

Begin

User Sign Up

Check Username availability

If username available

input other credentials

create user

else

re-enter username

End

Then Begin

Sign in

Check credentials

If Credentials match

Login User

Break;

Else

take back to Start Screen

End

8.2 Track Satellite

User is already logged in

Begin

Select getLocation (ajax)

Display Lats and Longs

Select satellite

Click the Track Button

Fetch satID

Fetch current TLE (GET method)

Generate SGP4 calculations

Generate czml file

Map the czml file on cesium clint

Display orbit

Enable / disable swath on click

Enable Alert on click

End

8.3 **Search**

User is already logged in

Begin

Select text field

Enter satName / satID / noradID

Click Search button

Query search database

Display results if available else error message

End

8.4 **Get User Location**

User is already Logged IN

Begin

Press getLocation button

Geolocation ajax invoked and last logs returned

Display lats and longs in the texts fields

Plot the location on the cesium client

End

APPENDICES

Appendix A: Glossary

1 SGP4

The Simplified General Perturbations (SGP) model series began development in the 1960s (Lane 1965), and became operational in the early 1970s (Lane and Cranford, 1969). A variety of publications presented the mathematical theory, and the paper by Vallado et al. (2006) tracks the history of these publications. Unfortunately, there has never been a release of any kind of differential correction code to implement the SGP4 method in a systematic approach to create Two-line Element (TLE) data. With the increased number of observing sites, and the availability of low-cost high quality optical observations, it is desirable to have such codes. The primary uses would be to obtain a more accurate TLE from independent data, and to have the ability to examine covariance data to support mission operations (such as conjunction operations). Spacetrack Report Number 3 (Hoots et al 1980) noted the importance of using the specific equations and data input to ensure proper operation and we repeat it here. “The most important point to be noted is that not just any prediction model will suffice… The NORAD element sets must be used with one of the models described in this report in order to retain maximum prediction accuracy.” This compatibility applies for the SGP4 propagation code, as well as the orbit determination code. We noted several minor points in the original SGP4 paper in which the performance of SGP4 could be improved. To maximize the usefulness of these features one should ideally use TLE formed with differential correction using an identical model as well. Thus, this paper provides a way to accomplish this action.

2 Cesium Maps

Cesium Maps is a web mapping service application and technology provided by Bing. It's an open-source JavaScript library for world-class 3D globes and mapsIIt offers street maps and a route planner for traveling. Cesium is an open source geospatial visualization JavaScript library tuned for dynamic display of data. The Cesium product line provides convenience products that enable your implementation of Cesium.

3 NORAD ID

The Satellite Catalog Number (also known as NORAD Catalog Number, NASA catalog number, USSPACECOM object number or simply Catalog number and similar variants) is a sequential 5-digit number assigned by United States Space Command to all Earth orbiting satellites in order of identification.

4 TLE

Two Line Elements is a format of two lines containing alpha numeric values in a segregated break down fashion which describes about many things regarding satellites including the satellite IDs and the *Kaplerian Elements* which help to determine the satellite orbit and motion as mentioned in the table below:-

Data for each satellite consists of these lines in the following format:

AAAAAAAAAAAAAAAAAAAAAA
1 NNNNNU NNNNNAAA NNNNN.NNNNNNNN +.NNNNNNNN +NNNNN-N +NNNNN-N N NNNNN
2 NNNNN NNN.NNNN NNN.NNNN NNNNNNN NNN.NNNN NNN.NNNN
NN.NNNNNNNNNNNNN

Line 1	
Column	**Description**
01	Line Number of Element Data
03-07	Satellite Number
08	Classification (U=Unclassified)

10-11	International Designator (Last two digits of launch year)
12-14	International Designator (Launch number of the year)
15-17	International Designator (Piece of the launch)
19-20	Epoch Year (Last two digits of year)
21-32	Epoch (Day of the year and fractional portion of the day)
34-43	First Time Derivative of the Mean Motion
45-52	Second Time Derivative of Mean Motion (decimal point assumed)
54-61	BSTAR drag term (decimal point assumed)
63	Ephemeris type
65-68	Element number
69	Checksum (Modulo 10) (Letters, blanks, periods, plus signs = 0; minus signs = 1)

Line 2	
Column	**Description**
01	Line Number of Element Data
03-07	Satellite Number
09-16	Inclination [Degrees]
18-25	Right Ascension of the Ascending Node [Degrees]
27-33	Eccentricity (decimal point assumed)
35-42	Argument of Perigee [Degrees]
44-51	Mean Anomaly [Degrees]
53-63	Mean Motion [Revs per day]
64-68	Revolution number at epoch [Revs]

69	Checksum (Modulo 10)

ANALYSIS & EVALUATION

1. Overview

The purpose of this chapter is to give the user a clear physical demonstration of the functionalities of SAS. This chapter is aimed to eliminate ambiguities and misunderstandings that may exist. It will explain all functions that the software should perform.

2. Test Cases

Test – case identifier	**TC-1**
Use Case Tested :	**UC-1(Cesium Viewer)**
Pass/Fail criteria :	**The test passes if cesium viewer appears in the UI page (after he is logged in , in the system)**
Input Data :	**Alpha Numeric**
Test Procedure :	**Results (Expected)**
1. Cesium Viewer appears	**The system will display the cesium globe in the UI page in the cesium container div**
2. Cesium Viewer doesn't appear	**The system will not display the cesium globe in the UI page in the cesium container div**

Test – case identifier	**TC-2**
Use Case Tested :	**UC-2(Track satellite)**
Pass/Fail criteria :	**The satellite orbit will load and display on demand in the cesium globe**
Input Data :	**Alpha Numeric**
Test Procedure :	**Results (Expected)**
1. Czml data loaded in the cesium viewer	**satellite orbit added and displayed in the cesium viewer onclick of respective satellite button**
2. Czml data not loaded in the cesium viewer	**satellite orbit added and displayed in the cesium viewer onclick of respective satellite button**

Test – case identifier	TC-3
Use Case Tested : Pass/Fail criteria : Input Data :	UC-3(Fullscreen Button) The Cesium viewer will add the Fullscreen button to its toolbar Alpha Numeric
Test Procedure :	Results (Expected)
1. The Fullscreen button added to the cesium viewer toolbar 2. The Fullscreen button not added to the cesium viewer toolbar	Cesium Viewer will display Fullscreen button Cesium Viewer will not display the Fullscreen button

Test – case identifier	TC-4
Use Case Tested : Pass/Fail criteria : Input Data :	UC-4(Show Swath Button) The Cesium viewer will add the Show Swath button to its toolbar Alpha Numeric
Test Procedure :	Results (Expected)
1. The Show Swath Button added to cesium viewer toolbar 2. The Show Swath Button not added to the cesium viewer toolbar	Cesium Viewer will display Show Swath Button Cesium Viewer will not display the Show Swath Button

Test – case identifier	TC-5
Use Case Tested : Pass/Fail criteria : Input Data :	UC-5(Fullscreen) The Cesium viewer will display in Fullscreen onclick of its button Alpha Numeric
Test Procedure :	Results (Expected)
1. The Fullscreen button functionality Executes on click 2. The Fullscreen button functionality doesn’t executes on click	Cesium Viewer will display in Fullscreen Cesium Viewer will not display in Fullscreen

Test – case identifier	TC-6
Use Case Tested : Pass/Fail criteria : Input Data :	UC-6(Show Swath) The Cesium viewer will add the new entity (cone)to the cesium viewer Alpha Numeric
Test Procedure :	Results (Expected)
1. The new entity (cone) added to the cesium viewer 2. The new entity (cone) not added to the cesium viewer	Cesium Viewer will display Swath Cesium Viewer will not display the Swath

Test – case identifier	TC-7
Use Case Tested : Pass/Fail criteria : Input Data :	UC-7(Search Satellite) The user may search a satellite by its id or name Alpha numeric
Test Procedure :	Results (Expected)
1. Match found 2. Match not found	1. The system will display the concerned data 2. The system will generate the "SATELLITE NOT FOUND " message

Test – case identifier	TC-8
Use Case Tested : Pass/Fail criteria : Input Data :	UC-8(Get Location) The cesium viewer will display a yellow dot at the user location and add the cartographic position to the Longitude and Latitude bars Alphanumeric
Test Procedure :	Results (Expected)
1. The user will click on the Get Location button and the success scenario executes 2. The user will click on the Get Location button and the failure scenario executes	1. The cesium viewer will display a yellow dot at the user location and add the cartographic position to the Longitude and Latitude bars 2. The error message will be displayed

Test – case identifier	TC-9
Use Case Tested :	UC-9(Set Location)

Pass/Fail criteria :	**The cesium viewer will display a red dot at the Cursor click location and add the cartographic position to the Longitude and Latitude bars**
Input Data :	**Alphanumeric**
Test Procedure :	**Results (Expected)**
1. The user will click on the Set Location button and the success scenario executes **2. The user will click on the Set Location button and the failure scenario executes**	**1. The cesium viewer will display a red dot at the Cursor click location and add the cartographic position to the Longitude and Latitude bars** **2. The cesium viewer will not display anything**

Test – case identifier	**TC-10**
Use Case Tested :	**UC-10(Reset)**
Pass/Fail criteria :	**The Cesium viewer will remove all the entities loaded on the cesium viewer**
Input Data :	**Alpha Numeric**
Test Procedure :	**Results (Expected)**
1. The user will click on the Reset button **2. The user will click on the Reset button**	**The Cesium viewer will remove all the entities loaded on the cesium viewer** **The Cesium viewer will not remove the entities loaded on the cesium viewer**

Test – case identifier	**TC-11**
Use Case Tested :	**UC-11(Get Alert)**
Pass/Fail criteria :	**Alert notifications will generate on the respective czml data loaded in the cesium viewer**
Input Data :	**Alphanumeric**
Test Procedure :	**Results (Expected)**

1. The user will click on the Get Alert button and then on the czml object (satellite) in the cesium viewer **2. The user will click on the Get Alert button and then on the czml object (satellite) in the cesium viewer**	**1. Alert notifications appear when the satellite is 200km short of user location and again when it hovers above the user location** **2. Alert notifications do not appear**

Test – case identifier	**TC-12**
Use Case Tested : **Pass/Fail criteria :** **Input Data :**	**UC-12(Register users menu)** **The test passes if the popup registration menu appears** **Alpha Numeric**
Test Procedure :	**Results (Expected)**
1. Menu appear **2. Menu doesn't appear**	**The system prompts the popup registration menu (at the right side of login menu)** **The system doesn't prompts the popup registration menu (at the right side of login menu)**

Test – case identifier	**TC-13**
Use Case Tested : **Pass/Fail criteria:** **Input Data :**	**UC-13(Register users)** **The test passes if the new user successfully sends his registration application.** **Alpha Numeric**
Test Procedure :	**Results (Expected)**
1. Message appear **2. Message doesn't appear**	**The system prompts that the user application for registration has been successfully sent** **The system doesn't prompt that the new user registration application has been sent**

FUTURE WORK

1. Overview

The purpose of this chapter is to elaborate what will the future aspects of this project in relation with the notion of its prospects in both civil use and military as the application has scope in both the circles.

2. Reusability

The modular design has been employed in order to leave a room for the future reusability and enhancement aspects of the modules. A simpler user interface of the system has been incorporated, so as to extend the usability of the system to all the novice users of the application.

The time has to be traded off for more enhanced features including the induction of measurement tools for ground analysis for decision making in different scenarios especially in the operational planning.

3. Future Prospects (Military)

At the successful completion of this project the notion is to deploy this application into the existing office automations system's portal of Pakistan Army to extend its uses to the existing military interests and requirements.

The ongoing manual paper based alerts and notifications of the hostile country satellites can be swapped away by a digital platform available 24/7 to the staff officers at all times along with its ease of use and better assimilation of the issue in context.

The system could be further of interest to the Pakistan Airforce, Pakistan Navy and specially to the Strategic Plans Division which can make use of its functionality in their daily tasks to better conceal themselves from the hostile satellite's surveillance.

4. Future Prospects (Civil)

The application could also be of interest to the Pakistan Space and Upper Atmosphere Research Commission (*SUPARCO*), the national space agency, as its can calculate the position of satellites at various and current time which can be very beneficial for the scientists / individual / armature astronomers interested in the observation or studies or various experimental and other uses both in their individual and national capacities.

5. Conclusion

The technological progress in the present world is the constant factor of chance in the existing civil and military technologies and in order to keep up with the rest of the world at its advancing pace one has to step up and this can only be achieved through the advancements of the existing procedures and infrastructures in both technical and tactical fields.

SAS provides this opportunity in one special area of interest. His could be mutually beneficial for both civil and military interests. Moreover, after deploying it on Office Automation System's Portal, it may be further developed for more efficient reporting and analysis of satellites and ground without changing its main functionality and can serve as a digital mechanism to better address the issue.

BIBLIOGRAPHY

1. Visit to C4I Team

C4I team was visited for the purpose of information acquisition and learning the existing system and compatibility of the project with it. Guidelines were noted and duly kept under note during the development process.

2. References

2.1 Info About The Kaplerian Elements. (n.d.). Retrieved November 5, 2016, from https://marine.rutgers.edu/cool/education/class/paul/orbits.html

2.2 Kaplerian Elements. (n.d.). Retrieved November 5, 2016, from http://www.amsat.org/amsat-new/tools/keps_detail.php

2.3 Satellite Orbits info. (n.d.). Retrieved November 5, 2016, from http://myreaders.info/05_Satellites_Orbit_Elements.pdf

2.4 Satellite Orbits info. (n.d.). Retrieved November 6, 2016, from http://www.n2yo.com/?s=25544

2.5 Satellite Position Data (TLE). (n.d.). Retrieved November 6, 2016, from http://celestrak.com/NORAD/elements/

2.6 Satellite Position Data (TLE) tracker. (n.d.). Retrieved November 6, 2016, from http://www.satellite-calculations.com/TLETracker/SatTracker.html

2.7 Master Index of Satellites. (n.d.). Retrieved November 6, 2016, from https://www.celestrak.com/NORAD/elements/master.asp

2.8 The United Nations Registry of Space Objects. (n.d.). Retrieved November 6, 2016, from http://www.planet4589.org/space/un/un_taba.html

2.9 Satellites Catalogs. (n.d.). Retrieved November 6, 2016, from http://celestrak.com/satcat/search.asp

2.10 Satellite movement using TLE. (n.d.). Retrieved November 6, 2016, from http://www.amostech.com/TechnicalPapers/2007/Modeling_Analysis_Simulation/Kelecy.pdf