

Verification and Validation Report:

March 7, 2024

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

symbol	description
ObjectID	MongoDB Record/Document Id
DB	Database
Schedule	Model type for MongoDB Schedule Collection
Command	Model type for MongoDB Command Collection

symbols, abbreviations or acronyms – you can reference the SRS tables if needed

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Functional Requirements Evaluation	1
4	Nonfunctional Requirements Evaluation	3
4.1	Usability	3
4.2	Performance	5
4.3	Environmental	5
4.4	Maintenance	6
4.5	Integrity	7
4.6	Privacy and Accessibility	8
4.7	etc.	9
5	Comparison to Existing Implementation	9
6	Unit Testing	9
6.1	Scheduling Module	9
6.1.1	API Endpoints	9
6.2	Satellite Users Module	13
6.2.1	API Endpoints	13
6.3	Helper Functions	19
7	Changes Due to Testing	22
8	Regression Testing	23
8.1	Authentication Module	23
8.2	Satellite Module	24
9	Automated Testing	24
10	Trace to Requirements	24
11	Trace to Modules	24
12	Code Coverage Metrics	25

List of Tables

List of Figures

1	A simple caption for the image.	6
---	---	---

This document ...

3 Functional Requirements Evaluation

FR-3 (The MCT will present a graphical interface allowing users to enter, select, and execute satellite commands while receiving text output from each command)

- The ability to enter satellite commands in the backend database has been tested with jest.
- The ability to execute satellite commands for a schedule in the backend has been tested with jest.
- The ability to enter satellite commands through the frontend has been tested manually.

FR-4 (The MCT will log all commands sent to and received from each target with timestamps.)

- The logging of data in the backend database have been tested through unit testing by emulating the execution of satellite commands.

FR-5 (The MCT must present a graphical user interface to create, edit, and delete automated command sequences)

- The addition, modification, and deletion of command sequences have been tested manually through the frontend.

FR-6 (The MCT must have an option to cancel an automated command sequence during execution.)

- Testing has not yet been implemented.

FR-11 (The MCT will be able to reject a user from executing a command unless that command matches the permission list criteria for the command target.)

- Unit tests have been implemented to schedule commands for users with both sufficient and insufficient permissions.

FR-12 (The MCT must be able to schedule commands and automated command sequences for future execution.)

- Unit tests have been implemented to schedule commands for an overpass date in the future.
- The scheduling of commands have been tested manually through the frontend.

FR-13 (The MCT shall present a graphical user interface to create, edit, and delete scheduled commands and sequences.)

- The addition, modification, and deletion of commands to a schedule have been tested manually through the frontend.

FR-14 (The MCT shall present a graphical user interface to select and edit the satellites of interest)

- Setting TLE data for users to choose have been tested with valid, invalid, and empty TLEs, to ensure the correct output is received.

FR-15 (The MCT shall display the current orbital state for satellites of interest, elevation, and solar illumination.)

- New changes to FR-SLN14 have been made to test the fetching of the data from external libraries (instead of testing for correctness). These tests includes fetching the data based on the state of the TLE passed in, Valid TLE, Invalid TLE, Valid Start Date, Invalid State Date, and a combination of both.
- For testing solar illumination of the satellite, testing includes fetching the data based off the validity of the date, and if the input units are valid (for the external library holder)
- For polar plot data, testing includes fetching the data based off the validity of the start and end date.
- Lastly, the return type of the backend api call is tested to ensure the data is returned in the proper format, with correct typing.

FR-16 (The MCT shall be able to schedule commands when a satellite enters a user-specified elevation threshold over the ground station coordinates.)

- Removed as per stakeholder requests .

FR-17 (The MCT shall be able to schedule commands when a satellite enters or exits in the covered area.)

- Unit tests have been implemented to ensure scheduled commands are only executed on the date and time range specified for an overpass date.

4 Nonfunctional Requirements Evaluation

4.1 Usability

usability-1 - NFR 10.1

- The feedback we gained for this test was combined with the second usability test which helps in testing NFR 10.3.

usability-2 - NFR 10.3

- The initial testing involved users with no prior experience with the application, focusing on their ability to learn and navigate the system independently. The feedback highlights both strengths and areas for enhancement in the application's design and functionality.
- Strengths Noted:
 - The satellite and overpass functionality received positive remarks for its quick response when navigating to specific longitudes and latitudes, indicating efficient performance in these areas.
 - The command deletion process within the application was described as "pretty solid," suggesting it is user-friendly and reliable.
- Areas for Improvement:
 - Visibility and Clarity: There was confusion about the "valid commands box," suggesting a need for increased visibility and clearer indications of its purpose. It's essential to make it more obvious to users what the valid commands are.

- Logs Page Functionality: Questions were raised about the purpose of the logs page, with suggestions to include filters for better usability, indicating a need for clearer presentation and functionality.
- User Interface (UI) Considerations:
 - * The all-white interface led to suggestions for a light mode/dark mode feature, catering to different user preferences.
 - * The addition of a feature to easily add satellites to the database directly from the first page was recommended.
 - * Faster loading times for satellite changes were requested to enhance user experience.
 - * Improvements in readability and presentation of information for upcoming satellite passes, possibly through a tabular format, were suggested to make the data more accessible.
 - * A reevaluation of navigation within the schedule queue, seeking alternatives to horizontal scrolling, was advised.
 - * The introduction of a filter for satellite passes, enabling searches by date or specific passes, was recommended to improve usability.
 - * A call for a design that avoids the need for page scrolling, aiming to present all relevant information simultaneously.
 - * The presentation of information in blocks or separated formats was suggested to improve visual hierarchy and readability, particularly for satellite information.

usability-3 - NFR 10.5

- After careful consideration, our team has decided not to proceed with the implementation of the accessibility-focused verification and validation (VnV) plan, as outlined under the Usability-3 category, NFR: 10.5. This decision reflects our current prioritization of resources and development efforts.
- While we recognize the importance of accessibility in software development, our decision to forego this specific aspect of testing at this time allows us to allocate our resources towards other critical areas of

development. We remain committed to creating an inclusive and accessible application and plan to revisit and incorporate comprehensive accessibility features and testing in future development phases.

4.2 Performance

performance-3 - NFR 11.3

- Precision calculation will be tested by external library providers as the calculations are no longer in the scope of the application.

performance-4 - NFR 11.4

- Scenarios which cause internal exceptions have been identified and implemented in automatic test cases, and is handled appropriately, be it error messages or additional error handling functionality.

4.3 Environmental

environmental-1 - NFR 12.2.1

- The application is setup to be hosted on Netlify's free service at the current moment. This applies that the features are also limited and that there cannot be a lot of traffic sent which makes testing the load much challenging. However, JMeter was used to verify the capabilities and challenged the application with 20 virtual users for 20 minutes. Here are the results:



Figure 1: A simple caption for the image.

environmental-2 - NFR 12.2.2

- To achieve this, we leveraged Netlify’s powerful development platform. Netlify facilitated the creation of a self-contained and fully functional development environment, streamlining the workflow and significantly enhancing productivity. The developers were able to set up their local environments effortlessly, following the instructions provided, and could launch the development setup with confidence, knowing it would initialize and run without errors.

environmental-3 - 12.5

- We are using GitHub issues to keep track of the dates and time for the project. Issues are created for the tasks that are remaining to be performed along with attaching the development branch that focuses on it. This way, we are able to ensure that the functional, non-functional and outside requirements are being fulfilled on-time.

4.4 Maintenance

maintenance-2 - NFR 13.1.2

- GitHub Actions is correctly configured for the project, triggering automated build and test checks. The test checks are triggered every commit, and every PR, seamlessly integrated in the Github repository.

maintenance-3 - NFR 13.1.3

- The system identifies an outdated version automatically by utilizing node's npm install command before the server is ran, and applies any changes (upgrades and downgrades). In addition, deliberate code formatting issues were introduced, and automatically corrected by esLint, the chosen code linter.

maintenance-4 - NFR 13.1.4

- The project directory structure is intuitive, code is divided into client and server folders (front-end and back-end), with components, styles, and pages being divided for logical ordering. Variable, function, and template names follow consistent naming schemes, and are grouped where needed. Lastly, rigorous code reviews are performed to maintain the quality of the repository.

maintenance-5 - NFR 13.1.5

- A code formatting tool, ESLint is integrated in the development environment through Visual Studio Code, and is enforced in code review. Since it is automatically integrated, random code files which are inspected match indentation, variable naming, and code structure.

4.5 Integrity

integrity-1 - NFR 14.2.1

- Testing the Overpass Calculation will be tested by external library providers as the calculations are no longer in the scope of the application.

integrity-2 - NFR 14.2.2

- Testing the Minimum Elevation will be tested by external library providers as the calculations are no longer in the scope of the application.

integrity-5 - NFR 14.2.5

- Various simulated satellite failures have been implemented in test cases to test and monitor error messages which are consistent amongst other messages, and communicate the nature of missing data attributes.

integrity-6 - NFR 14.2.6

- The application performs essential functionalities on all browsers (Safari, Chrome, Firefox).

integrity-7 - NFR 14.2.7

- The typescript compiler verifies HTML and CSS validity, and has not identified any compatibility issues across devices. In addition, esLint is applied to apply HTML and CSS coding standards when it comes to formatting. The application user interface remains consistent and functional across different devices and browsers.

4.6 Privacy and Accessibility

access-1 - NFR 14.1.1

- The application integrates with an external authentication service (Auth0) to manage user authentication and account management. New and existing users will be registered and logged in using this service. The test cases for this requirement are AM2 and AM3 and it can be found below in section 8 of Regression Testing.

access-2 - NFR 14.1.2

- The application integrates with an external authentication service (Auth0) to manage user authentication and account management. This service allows an administrator to configure a rate limit of 10 login attempts per minute. As a result, the application enforces a 1 minute timeout period before the user can attempt to authenticate again. The test case for requirement is AM1 and it can be found below in section 8 of Regression Testing.

privacy-1 - NFR 14.3.1

- The test case for this requirement has not been implemented yet. Refer to test case REG-SAT1 in section 8 of Regression Testing.

4.7 etc.

5 Comparison to Existing Implementation

This section will not be appropriate for every project.

6 Unit Testing

6.1 Scheduling Module

6.1.1 API Endpoints

POST /schedule/createScheduledCommand

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM1	FR-12, FR-13, FR-17, FR-16	userId: ObjectID, satelliteId: ObjectID, com- mandId: ObjectID, command: "start"	userId: ObjectID, satelliteId: ObjectID, com- mandId: ObjectID, command: "start"	userId: ObjectID, satelliteId: ObjectID, com- mandId: ObjectID, command: "start"	Pass
SM2	FR-17, FR-16, FR-13	userId: ObjectID, satelliteId: ObjectID, com- mandId: ObjectID, command: "com- mandNot- InCriteria"	status: 500, Error: "Invalid command sequence or user permis- sions"	status: 500, Error: "Invalid command sequence or user permis- sions"	Pass

PATCH /schedule/updateScheduledCommand

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM3	FR-3, FR-4, FR-5	userId: ObjectID, satelliteId: ObjectID, commandId: ObjectID, command: "start"	userId: ObjectID, satelliteId: ObjectID, commandId: ObjectID, command: "start"	userId: ObjectID, satelliteId: ObjectID, commandId: ObjectID, command: "start"	Pass
SM4	FR-11	userId: ObjectID, satelliteId: ObjectID, commandId: ObjectID, command: "commandNotInCriteria"	status: 500, Error: "Invalid command sequence or user permissions"	status: 500, Error: "Invalid command sequence or user permissions"	Pass

GET /schedule/getSchedulesBySatellite

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM5	FR-13	satelliteId: ObjectID	Message: "Fetched schedules", schedules: Schedule[]	Message: "Fetched schedules", schedules: Schedule[]	Pass
SM6	FR-13	satelliteId: ObjectID, status: "PASSED"	Message: "Fetched schedules", schedules: Schedule[]	Message: "Fetched schedules", schedules: Schedule[]	Pass

GET /schedule/getCommandsBySchedule

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM7	FR-13	satelliteId: ObjectID	Message: “Fetched com- mands”, com- mands: Com- mand[]	Message: “Fetched com- mands”, com- mands: Com- mand[]	Pass

DELETE /schedule/deleteScheduledCommand

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM8	FR-13	commandId: ObjectID, userId: ObjectID	Message: “Success. Deleted schedule”	Message: “Success. Deleted schedule”	Pass
SM9	FR-11	commandId: ObjectID, userId: ObjectID	status: 500, Error: “Insuf- ficient permis- sions”	status: 500, Error: “Insuf- ficient permis- sions”	Pass

GET /satellite/getSatelliteInfo

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM10	FR-15		status: 200	status: 200	Pass

GET /getPolarPlotData

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM11	FR-15	startTime = 2024- 01- 06T 10:15:00Z; endTime = 2024- 01- 06T 10:22:00Z;	status: 200	status: 200	Pass
SM12	FR-15	startTime = 2024- 01- 06T 10:15:00Z; endTime = "";	status: 500	status: 500	Pass

GET /maxElevation

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM13	FR-16	startTime = 2024- 01- 06T 10:15:00Z; endTime = 2024- 01- 06T 10:22:00Z;	status: 200	status: 200	Pass
SM14	FR-16	startTime = 2024- 01- 06T 10:15:00Z; endTime = "";	status: 500	status: 500	Pass

GET /getNextPasses

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM15	FR-15	TLE = 59909	status: 200	status: 200	Pass
SM16	FR-15	TLE = ""	status: 500	status: 500	Pass

GET /getSolarIllumination

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM16	FR-15	TLE = 59909	status: 200	status: 200	Pass
SM17	FR-15	TLE = ""	status: 500	status: 500	Pass

POST /changeTLE

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM18	FR-14	TLE = 59909	status: 200	status: 200	Pass
SM19	FR-14	TLE = ""	status: 500	status: 500	Pass

6.2 Satellite Users Module

6.2.1 API Endpoints

POST /satelliteUser/createSatelliteUser

Id	Input	Expected Output	Actual Output	Result
SUM1	userId: ObjectID, satelliteId: ObjectID, adminId: ObjectID, validCommands: ["teardown"]	satelliteUserId: ObjectID, satelliteId: ObjectID, validCommands: ["teardown"], adminId: ObjectID	satelliteUserId: ObjectID, satelliteId: ObjectID, adminId: ObjectID, validCommands: ["teardown"]	Pass
SUM2	userId: ObjectID, satelliteId: ObjectID, adminId: "invalidAdminId", validCommands: ["teardown"]	status: 500, Error: "Invalid command sequence or user permissions"	status: 500, Error: "Invalid command sequence or user permissions"	Pass
SUM3	userId: ObjectID, satelliteId: "invalidSatelliteId", adminId: ObjectID, validCommands: ["teardown"]	status: 500, Error: "Invalid Ids"	status: 500, Error: "Invalid Ids"	Pass
SUM4	userId: ObjectID, satelliteId: ObjectID, adminId: ObjectID, validCommands: ["invalidCommand"]	status: 500, Error: "Invalid command sequence or user permissions"	status: 500, Error: "Invalid command sequence or user permissions"	Pass
SUM5	userId: "invalidUserId", satelliteId: ObjectID, adminId: ObjectID, validCommands: ["teardown"]	status: 500, Error: "Invalid command sequence or user permissions"	status: 500, Error: "Invalid command sequence or user permissions"	Pass

PATCH /satelliteUser/updateByUser

Id	Input	Expected Output	Actual Output	Result
SUM6	satelliteUserId: ObjectID, satelliteId: Ob- jectID, adminId: ObjectID, valid- Commands: [“teardown”]	satelliteUserId: ObjectID, satelliteId: Ob- jectID, adminId: ObjectID, validCom- mands: [”tear- down”], adminId: ObjectID	satelliteUserId: ObjectID, satelliteId: Ob- jectID, adminId: ObjectID, validCom- mands: [“tear- down”]	Pass
SUM7	satelliteUserId: ObjectID, satelliteId: ObjectID, adminId: ”in- validAdminId”, validCommands: [“teardown”]	status: 500, Error: ”Invalid command sequence or user permis- sions”	status: 500, Error: ”Invalid command sequence or user permis- sions”	Pass
SUM8	satelliteUserId: ObjectID, satelliteId: ”invalidSatel- liteId”, adminId: ObjectID, valid- Commands: [“teardown”]	status: 500, Error: ”Invalid Ids”	status: 500, Error: ”Invalid Ids”	Pass
SUM9	satelliteUserId: ObjectID, satelliteId: Ob- jectID, adminId: ObjectID, valid- Commands: [“invalidCom- mand”]	status: 500, Error: ”Invalid command sequence or user permis- sions”	status: 500, Error: ”Invalid command sequence or user permis- sions”	Pass
SUM10	satelliteUserId: ”invalidSatel- liteUserId”, satelliteId: Ob- jectID, adminId: ObjectID, valid- Commands: [“teardown”]	status: 500, Error: ”Invalid command sequence or user permis- sions”	status: 500, Error: ”Invalid command sequence or user permis- sions”	Pass

GET /satelliteUser/getUserBySatellite

Id	Input	Expected Output	Actual Output	Result
SUM11	satelliteId: ObjectID	Message: "Fetched satellite users", satelliteUsers: satelliteUsers[]	Message: "Fetched satellite users", satelliteUsers: satelliteUsers[]	Pass
SUM12	satelliteId: "invalidSatelliteId"	Message: "Fetched satellite Users", satelliteUsers: []	Message: "Fetched satellite Users", satelliteUsers: []	Pass

GET /satelliteUser/getCommandsBySatelliteAndUser

Id	Input	Expected Output	Actual Output	Result
SUM13	satelliteId: ObjectID, userId: ObjectID	Message: "Fetched satellite users", satelliteUsers: []	Message: "Fetched satellite users", satelliteUsers: []	Pass
SUM14	satelliteId: "invalidSatelliteId", userId: ObjectID	Message: "Fetched satellite Users", satelliteUsers: []	Message: "Fetched satellite Users", satelliteUsers: []	Pass
SUM15	satelliteId: ObjectID, userId: "InvalidUserId"	Message: "Fetched satellite Users", satelliteUsers: []	Message: "Fetched satellite Users", satelliteUsers: []	Pass

DELETE /satelliteUser/deleteByUser

Id	Input	Expected Output	Actual Output	Result
SUM16	adminId: ObjectID, satelliteUserId: ObjectID	Message: "Removed User from satellite"	Message: "Removed User from satellite"	Pass
SUM17	adminId: "invalidAdminId", satelliteUserId: ObjectID	Message: "Invalid Ids"	Message: "Invalid Ids"	Pass
SUM18	satelliteUserId: "invalidId", adminId: ObjectID	Message: "Invalid Ids"	Message: "Invalid Ids"	Pass

6.3 Helper Functions

executeScheduledCommands(satelliteId: ObjectID, scheduleId: ObjectID)
 \Rightarrow void

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM20	FR-10	satelliteId: ObjectID, scheduleId: ObjectID	List of log records corresponding to the executed command records for the specified scheduleId	List of log records corresponding to the executed command records for the specified scheduleId	Pass

rescheduleLeftoverCommand(satelliteId: ObjectID, scheduleId: ObjectID) \Rightarrow void

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM21	FR-10, FR-11	satelliteId: ObjectID, scheduleId: ObjectID	Schedule specified in the request has no commands with status: status: "QUEUED"	Schedule specified in the request has no commands with status: status: "QUEUED"	Pass

addSchedulesForNext7Days(satelliteId: ObjectID, noradId: number) ⇒
void

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM22	FR-16, FR-17	satelliteId: ObjectID, noradId: number	Satellite specified in the request has new schedules for the next seven days	Satellite specified in the request has new schedules for the next seven days	Pass

getSatelliteInfo(date: Date, tleLine1: number, tleLine2: number) ⇒ {
positionEci: number, velocityEci: number, longitude: number, latitude:
number, height: number, azimuth: number, elevation: number, rangeSat:
number }

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM23	FR-15	new Date(), 1 55098U 23001CT 23359.66872105 .00021921 00000-0 89042-3 0 9991, 2 55098 97.4576 58.0973 0014812 57.5063 302.7604 15.24489013 54199	{ posi- tionEci: number, veloci- tyEci: number, longitude: number, latitude: number, height: number, azimuth: number, elevation: number, rangeSat: number }	{ posi- tionEci: number, veloci- tyEci: number, longitude: number, latitude: number, height: number, azimuth: number, elevation: number, rangeSat: number }	Pass
SM24	FR-15	””, 1 55098U 23001CT 23359.66872105 .00021921 00000-0 89042-3 0 9991, 2 55098 97.4576 58.0973 0014812 57.5063 302.7604 15.24489013 54199	Error: In- valid Date	Error: In- valid Date	Pass
SM25	FR-15	new Date(), ””, ”” , ””	Error: In- valid TLE 21	Invalid TLE	Pass

isSunLit(date: Date, lon: number, lat: number, height: number) \Rightarrow boolean

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM26	FR-15	new Date(), 0, 0, 0	isDefined	isDefined	Pass
SM27	FR-15	"" , 0, 0, 0	isNotDefined	isNotDefined	Pass
SM28	FR-15	new Date(), 0, 0, 20000	Error: Height must be in km	Error: Height must be in km	Pass

setTLE(tle: string) \Rightarrow void

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
SM29	FR-14	"55098"	resolves	resolves	Pass
SM30	FR-14	"abcd"	Error: Invalid TLE	Error: Invalid TLE	Pass
SM30	FR-14	""	Error: Empty TLE	Error: Empty TLE	Pass

7 Changes Due to Testing

This section should highlight how feedback from the users and from the supervisor (when one exists) shaped the final product. In particular the feedback from the Rev 0 demo to the supervisor (or to potential users) should be highlighted.

FR-SLN15

- This test has been removed as the stakeholders no longer requested this requirement.

performance-3 (NFR 11.3)

- Precision calculation will be tested by external library providers as the calculations are no longer in the scope of the application.

access-1 (NFR 14.1.1)

- NFR 14.1.1 has been revised and the application will only require to support Single-Sign-On (SSO) authentication instead of Multi-Factor-Authentication (MFA).

8 Regression Testing

8.1 Authentication Module

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
AM1	NFR 14.1.2	email: "test1@gmail.com" password: "correct" Repeat 10 times	Disable login functionality for one minute	Disable login functionality for one minute	Pass
AM2	NFR 14.1.1	email: "test1@gmail.com" password: "correct"	Successfully log user into the application's home page	Successfully log user into the application's home page	Pass
AM3	NFR 14.1.1, NFR 14.1.2	email: "test1@gmail.com" password: "correct"	Successfully register user into the application and directs them to the home page	Successfully register user into the application and directs them to the home page	Pass

8.2 Satellite Module

Id	Reference Req. Id	Input	Expected Output	Actual Output	Result
REG-SAT1	NFR 14.3.1	port: number	Socket successfully connected with hash output	Socket successfully connected with hash output	TBD

9 Automated Testing

10 Trace to Requirements

Req. ID	System Test ID
FR-3	SM3
FR-4	SM3
FR-5	SM3
FR-10	SM10
FR-11	SM4, SM9, SM11
FR-12	SM1
FR-13	SM5, SM6, SM7, SM8
FR-16	SM1, SM2, SM12
FR-17	SM1, SM2, SM12
NFR-14.1.2	AM1
NFR-14.1.1	AM2, AM3
NFR-14.1.2	AM3
NFR-14.3.1	REG-SAT1

11 Trace to Modules

Module	System Test ID
Authentication Module	AM1, AM2, AM3
Schedule Module	SM1, SM2, SM3, SM4, SM5, SM6, SM7, SM8, SM9, SM10, SM11, SM12
Satellite User Module	SUM1, SUM2, SUM3, SUM4, SUM5, SUM6, SUM7, SUM8, SUM9, SUM10, SUM11, SUM12, SUM13, SUM14, SUM15, SUM16, SUM17, SUM18
Satellite Module	REG-SAT1

12 Code Coverage Metrics

References

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Reflection. Please answer the following question:

1. In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)
- Quinn - The biggest difference in the way that the VnV plan was different from what was conducted was for me, the satellite calculations. the initial plan was to develop the code to calculate, however upon future research, was determined to be more accurate to rely on scientists and utilize open-source libraries instead. As a result, the VnV plan was modified to instead validate the flow of data, specifically the data and types of the parameters. This could have been anticipated earlier with given research, and more effort will be given in the future by researching before establishing a plan for testing.
 - Umang - In terms of verifying and validating the functionality of the scheduling and authentication modules, we initially had wanted to automate both backend and frontend facing features using a unit-testing library. However, we instead decided to modularize the evaluation of the backend and frontend components. As a result, we opted for evaluating the backend API endpoints and helper functions using unit tests to measure the application's correctness. Then, we used acceptance testing with our stakeholders to assess the user interface of the application. Furthermore, this could have been anticipated by researching into common methods of testing server and client-side applications beforehand.
 - Diamond - Reflecting on our initial Verification and Validation (VnV) Plan, we realized the necessity to deviate from certain aspects, notably

the Usability-3, NFR: 10.5, focused on assessing accessibility. This part of the plan, aimed at manually testing the application's accessibility for users with diverse needs, was ambitious in ensuring our system was universally accessible. However, due to constraints in resources, time, and the availability of a varied group of test participants with specific accessibility needs, we have decided not to proceed with this detailed testing approach. Instead, our efforts have pivoted towards leveraging best practices in accessibility design and relying on automated tools to assess some accessibility features. This decision underlines the often complex balance between ideal testing scenarios and the practical limitations of project environments. It also highlights the importance of flexibility in project management and the need to adapt plans based on available resources and evolving project dynamics. This experience reinforces the value of incorporating adaptability into our VnV strategies, allowing us to better anticipate and accommodate changes in future projects while still aiming to deliver high-quality, accessible software within our constraints.

- Rishi - When we look back at our initial plan for Verification and Validation (VnV) and compare it to what we actually ended up doing, we noticed quite a few differences. These changes were necessary because we encountered unexpected requirements and had to make practical adjustments to fit the scope and methods of our project. For example, we had to tweak our strategies to fit new tools and technologies, adjust to changes in our team's skills, and cope with shifts in project deadlines.

These adjustments were all part of the unpredictable nature of software development. Plans that look good on paper often have to be adapted to deal with the real-world challenges we face, like technical hurdles, limited resources, and changes in project goals. Going through this process taught us the importance of being flexible and able to change direction when needed. While it's hard to predict every twist and turn, this project showed us that being adaptable is key to success.

We learned that having a solid plan is essential, but so is being ready to adjust that plan as we go along and learn new things. This mindset will help us anticipate and handle changes better in future projects, making sure our VnV activities stay on track with what we're trying

to achieve and the limitations we face.

- Dhruv - The biggest difference in our initial plan and what ended up happening was the new changes and requirements that were added along the way. As a result, practical changes are a consequence especially if we want to fit it within our timeline and scope of our project. This is only natural in software development and was an important lesson to learn because no one can ever predict all incoming changes that will happen. This forced me to not get fixated on one solution and to keep thinking from different perspectives. Additionally, we were able to do user testing with our stakeholders which provided important and useful feedback. This motivated us to research more software testing methods, even the unconventional ones perhaps. As someone who was primarily working on the front-end, I never estimated that it would take as long as it did. It was an important lesson that every aspect of software development is vital.