Daniela Muniz
Austin Douglas Crow
Justin Phillips
Max McKinney

Project Group 22

Self-Assembling Off-World Colony

Requirements Definition

Functional Requirements

- The system shall authenticate the supervisor’s credentials before access.
- The system shall display a list of controls for rovers, communication and assembler, along with options for the map, construction and diagnostics.
- The system shall display diagnostics, rover, satellite and assembly colony reports.
- The system shall display what it has carried out with the rovers and the assembly colony.
- The system shall report and display the rovers’ location, data, and action information.
- The system shall report and display the assembly colony’s location and data on environmental and colony needs.
- The system shall display its connection strength with the communication satellite.
- The system shall alert supervisor when there is a weak connection with the communication satellite.
- The system shall display the option for the supervisor to override the system itself in cases of emergency.
- The system shall display the the list of components that have been assembled on the colony.
- The system shall record and keep track of assembled components.
- The system shall display base, rover and assembly colony diagnostics.
- The system shall display life support.
- The system shall display when communications cannot happen and will queue a default operation until communications are reestablished.

Non-functional Requirements

- The system shall authenticate the supervisor within 10 seconds.
- The system shall update and display the rovers' location, data and action information as quickly as possible.
- The system shall update and display the assembly colony's location and data on environmental and colony needs as quickly as possible.
- The system shall update and report on its connection strength with the communication satellite as quickly as possible.
- The system shall display the the list of components that have been assembled on the colony from within the last minute.
- The system shall display diagnostics ran on the assembly colony from within the last minute.
- The system shall create and display reports when requested by the supervisor within 5 seconds.

Use Cases

Use Case 1: Aquatic Planet Operation

Actors:
- The self assembling colony
- Home base - Human Supervisor
- Waterbound Rovers
- Satellite

Pre-Conditions:
- Capsule has landed on the planet on the water's surface.
- Capsule has all necessary components to start the assembly process.
- Colony has established communications with the base on Earth so it can notify the home base that it has landed safely and can begin building the colony.

Postconditions:
- Capsule has been fully established and operational on the water's surface.
- The colony has been sectioned into 'tiles' where each tile contributes something to the colony, including tiles for:
  - Growing hydroponic crops
  - Processing potable water
  - Generating renewable energy
  - Communicating with the home base
  - Living quarters for humans
- Each section of the colony is afloat and can sustain human life

Flow of Events:
- Capsule disperses drones to survey the environment.
- If the surrounding area is suitable for a colony, capsule lands and begins construction.
- Colony deploys buoyant platforms with which to build on.
- Colony sections itself off into specialized buoyant tiles with unique functions.
- Colony communicates with the home base during assembly, updating on progress.
- If communication is interrupted, continues with construction independent of supervisor.
- Colony notifies home base upon assembly completion.
- Colony begins to perform regular maintenance and tests to keep it afloat and liveable.
- Home base sends inhabitants to the colony.

Use Case 2: Long Day/Night Cycle Planet

Actors:

- Supervisor
- Rovers
- Self-Assembler
- Satellites

Preconditions:

- User is on a hospitable habitat, in a location suitable for construction.
- Colony's planet has extremely long days and nights (e.g. Venus, where one day = 243 Earth days.)
- User has contact with assembler via satellites.
- User can perform the construction using the software, and is properly trained to do so.

Postconditions:

- Self Assembler has built the colony
- Colony has been equipped with power, water and breathable air
- Colony has been modified to handle the long nights
- Colony is ready for colonists

Flow of events:

- Self-assembler is sent to the planet to colonize
- Surface location is scouted by rovers to find the most optimal building location
- Self-assembler lands on that location
- Supervisor has established contact with the self-assembler, and oversees the process.
- Assembler starts the first steps in the AI process, the AI process is a set of steps the assembler follows while not under the control of the supervisor
- In the long blackout periods, the assembler will rely on its AI to continue assembly.
- Assembler builds the outer shell and the inside is now safe from outside weather
- Assembler now sets up the renewable power supply since nights are long it will rely on geothermal power, wind power and hydroelectric power
- Assembler sets breathable air system and clean water
- Supervisor performs safety check via the rovers and checks the safety gages to make sure everything is within the proper safety guidelines
- Colonists land on the colonized planet and settle into their new home

Use Case 3: Solar Flare Response

Actors:

- Human supervisor
- Rovers
- Self-Assembler System
- Satellite

Preconditions:

- Solar flare has damaged the satellite.
- System communication is lost between system and assembly colony.
- System communication is lost between system and rovers.
- Supervisor sees queue for default operation until communications are reestablished.

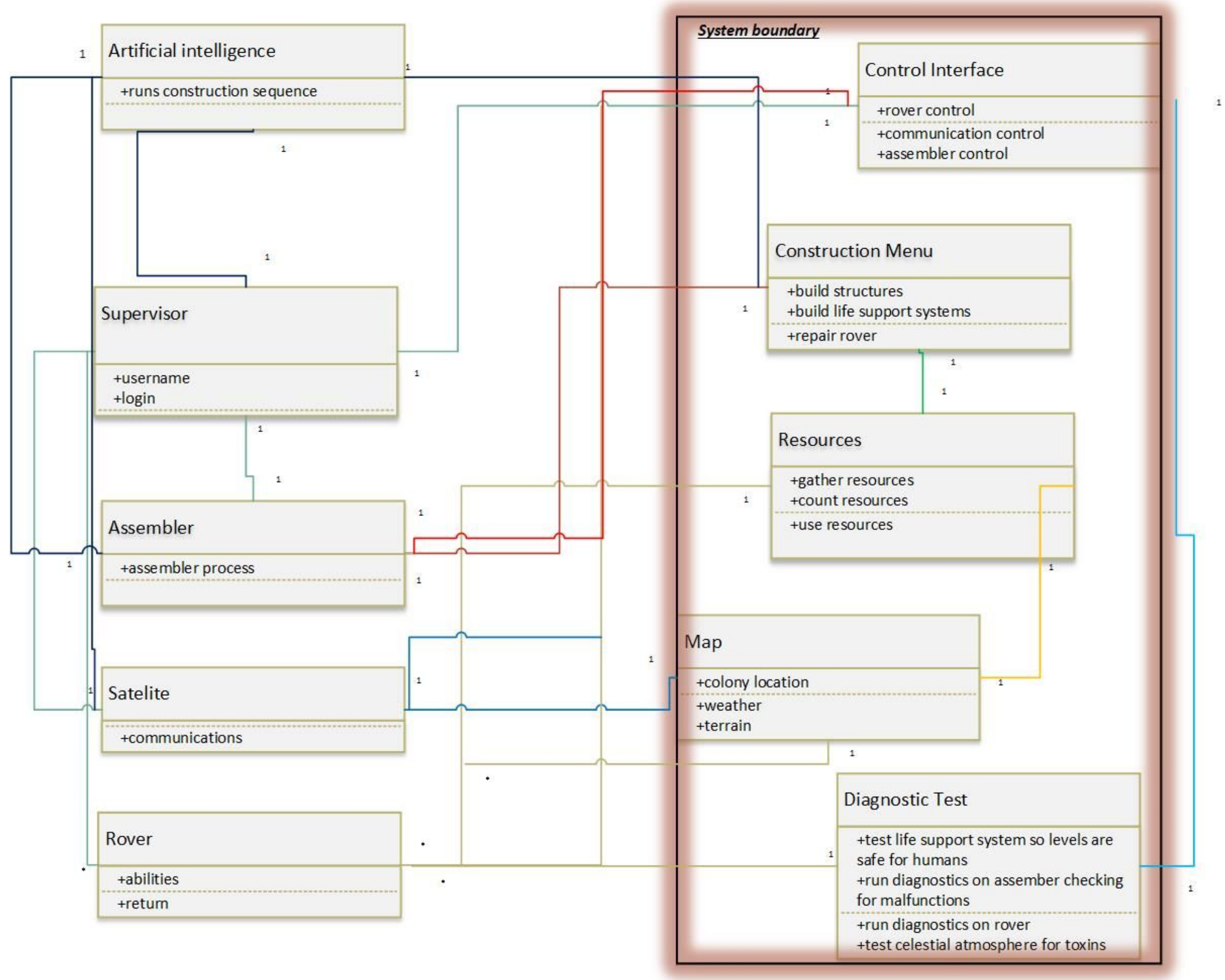
Postconditions:

- Supervisor has sent help to get system communications reestablished as soon as possible.
- Assembly colony and rovers operate on their own while action is taken to reestablish system communications.
- Replacement satellite and communication between rovers, assembly colony and system has been restored.

Flow of events:

- Supervisor sees queue for default operation.
- Supervisor sends satellite replacement request for system communication to get reestablished.
- Rovers and assembly colony go into default operation, independent from system.
- Supervisor remains observant of rovers and assembly colony.
- Replacement satellite is sent.
- Communication between rovers, assembly colony and system has been restored.

UML Class Diagram


System boundary
Artificial intelligence
+runs construction sequence
Control Interface
+rover control
+communication control
+assembler control
Construction Menu
+build structures
+build life support systems
+repair rover
Supervisor
+username
+login
Resources
+gather resources
+count resources
+use resources
Assembler
+assembler process
Map
+colony location
+weather
+terrain
Satelite
+communications
Diagnostic Test
+test life support system so levels are safe for humans
+run diagnostics on assember checking for malfunctions
+run diagnostics on rover
+test celestial atmosphere for toxins
Rover
+abilities
+return

Requirements Specification

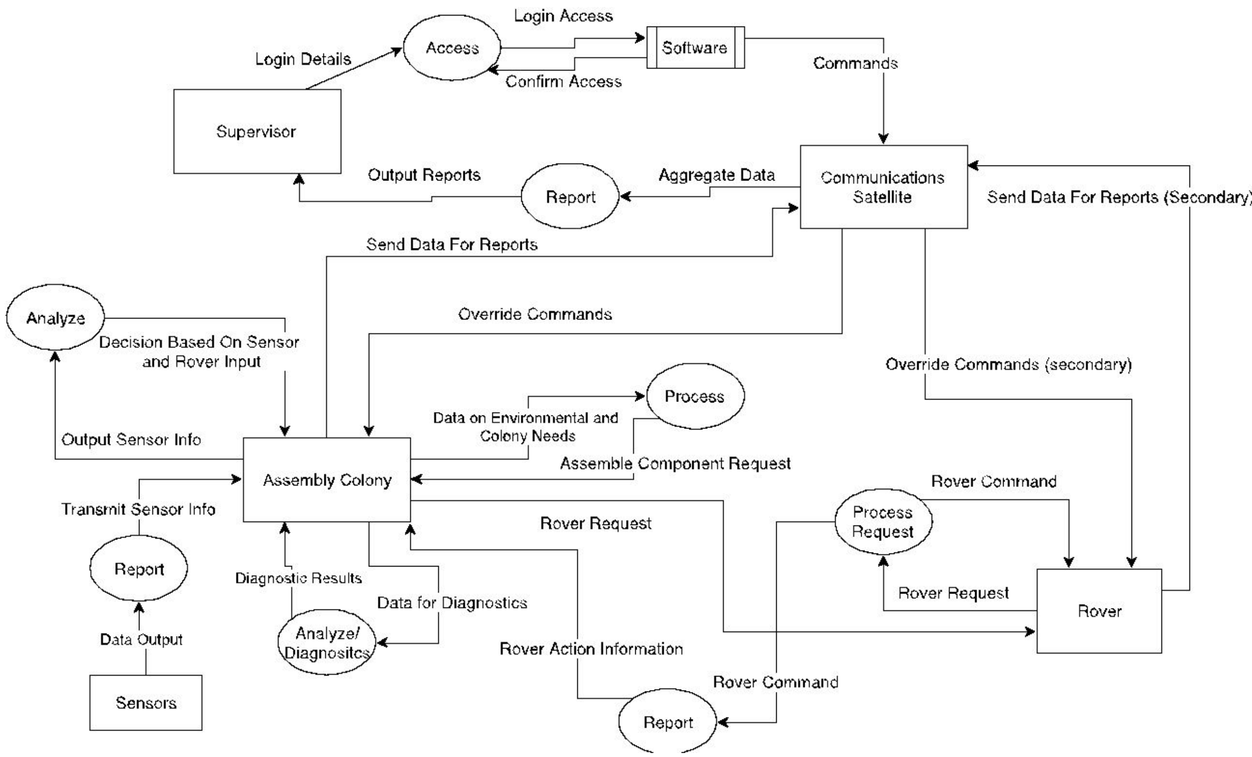
Functional Requirements

- The supervisor login details will be checked against the information of allowed users on the software's database.
- After the supervisor's login input succeeds, the system will display the list of commands.
- The system will autonomously direct the rovers and assembly colony.
- The system will communicate with the rovers and assembly colony via the communication satellite.
- The system's software will detect the rovers' and assembly colony's location, plot the locations, and graphically present it for the supervisor's readability on the interface.
- The system's software will detect and log the rovers' data and action information, parse the log, and translate the log into a presentable display for the supervisor.
- The system's software will detect and log the assembly colony's data on environmental and colony needs, parse the log, and translate the log into a presentable display for the supervisor.
- The system will run diagnostics on the assembly colony.
- The system will detect its connection with the communication satellite.
- When the supervisor selects command to override, the system will give up control and allow the supervisor to make decisions for the rovers and assembly colony.

Non-functional Requirements

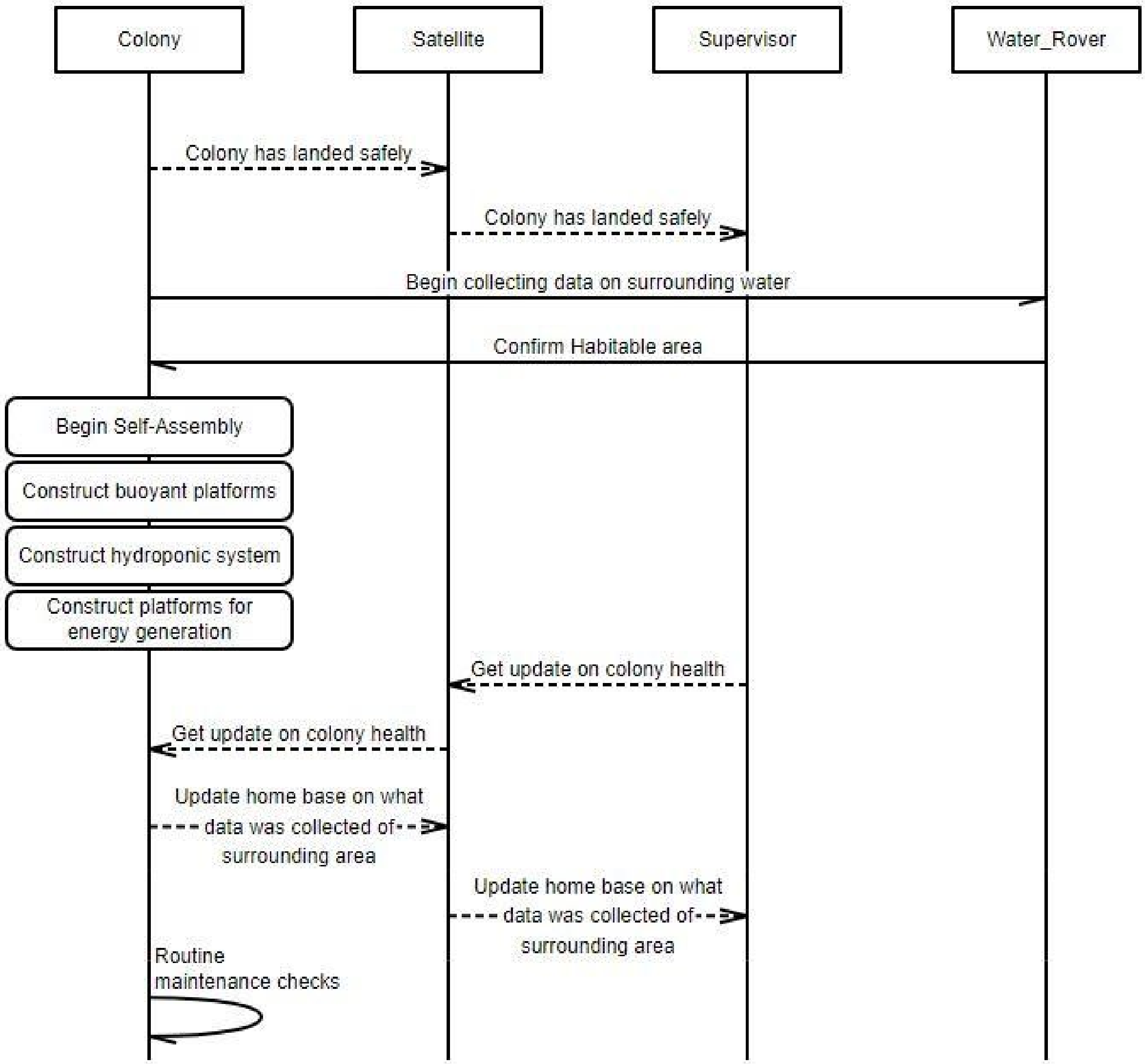
- The system will query the database and return a user authentication response in less than 10 seconds.
- The system will update the rovers' and assembly colony's data every second.
- The system's software will detect and log the assembly colony's data on environmental and colony needs, parse the log, and translate the log into a presentable display for the supervisor.
- The system will run diagnostics on the assembly colony every minute.
- The system will create reports in at most 5 seconds.
- The system will detect its connection with the communication satellite every second.

Data Flow Diagram

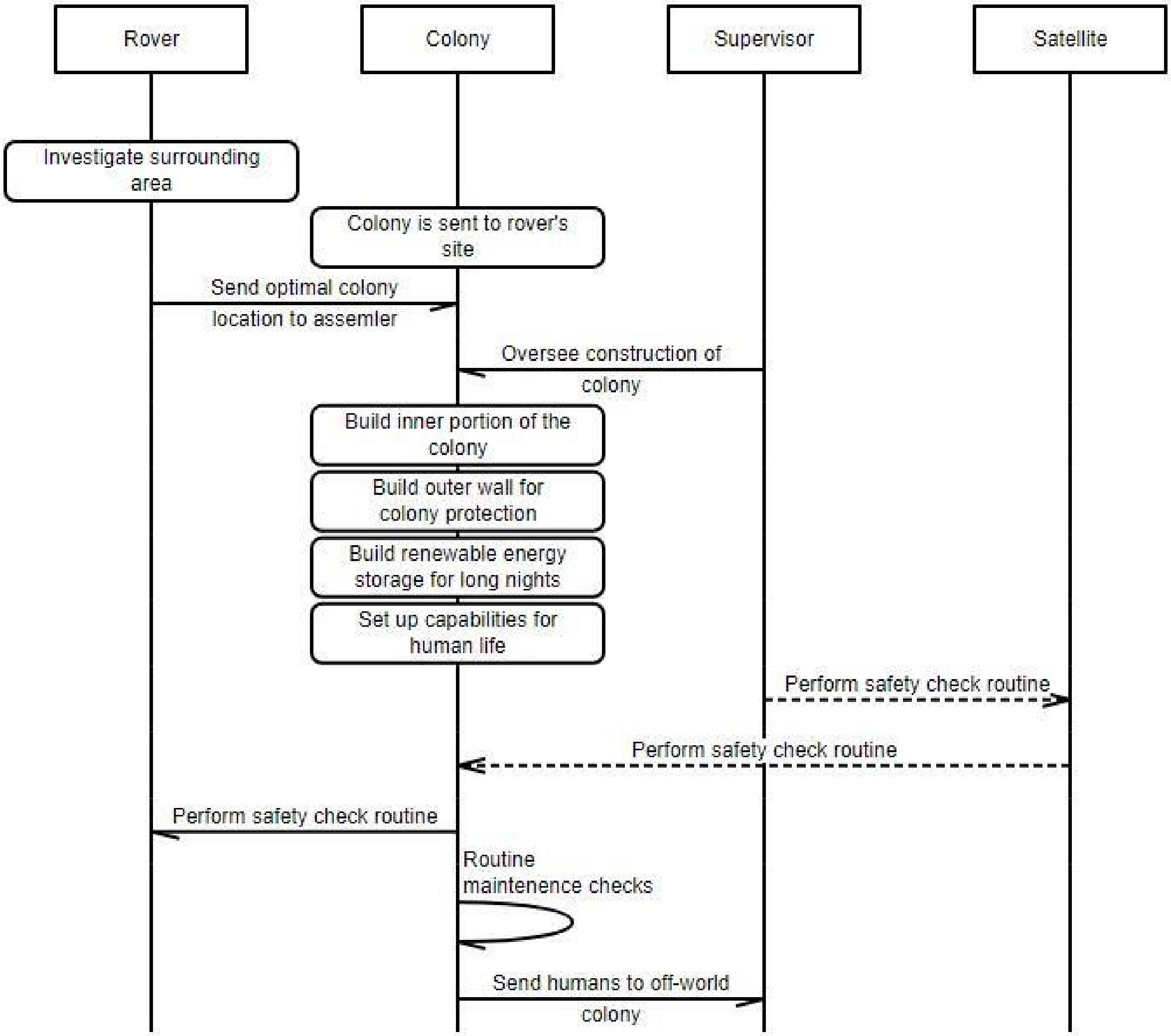


State Charts for Example Use Cases

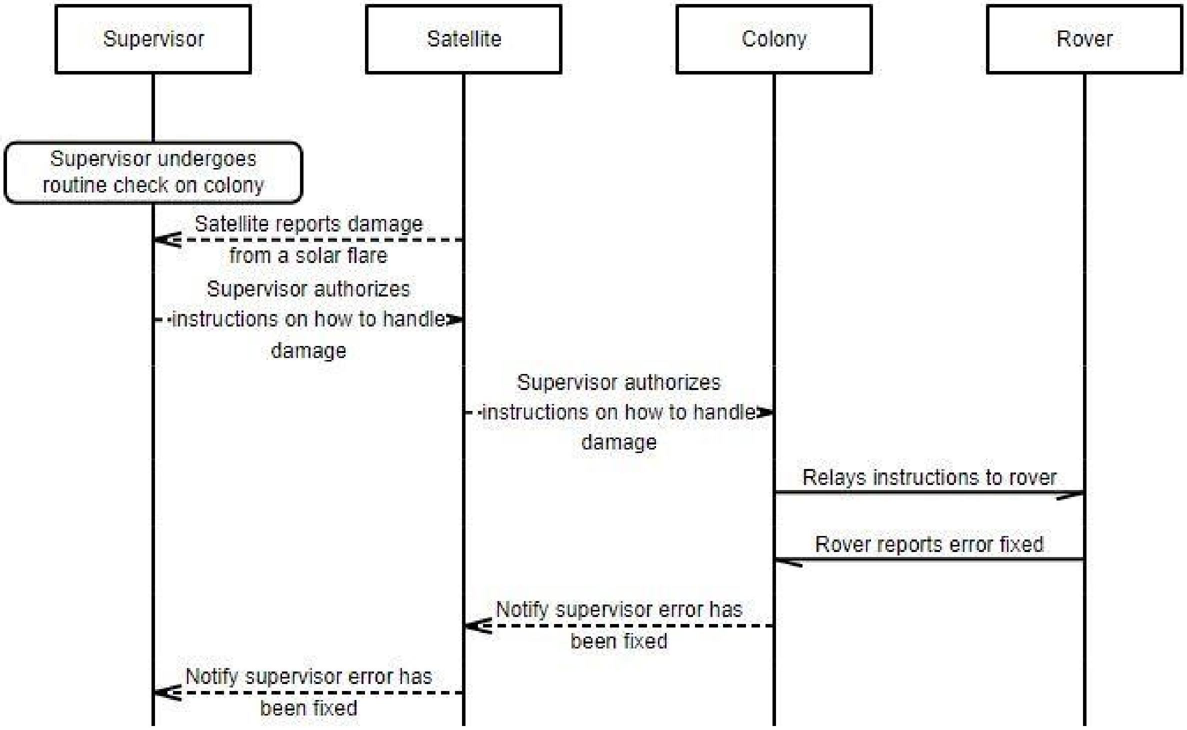
Case 1: Water-Bound Colony



Case 2: Long Day/Night Cycle Planet



Case 3: Solar Flare Response



<u>Changes Made to Requirements</u>

Some of the changes that were requested with Use Case 2 involved the AI portion of the system, and how it is responsible for the majority of the building process. Since the day cycle is so long, there are large swathes of time where the satellite may not communicate with the colony. As well, there will not be a real time connection with the supervisor since the distance between the two is so large. Because of this, the colony needs to have an AI that can handle portions of the construction on its own. In Use Case 2 we added that the AI will have a sequential building process that it will follow, if and when its not getting direct instructions from the supervisor also it was added that there will be an extended blackout period with supervisor since the rotation of the celestial body was extremely slow.

Use Case 1 remains mostly unchanged, as the customer was mostly pleased with it. He did, however, request that our Flow of Events have some more detail, so some entries were added. The most significant of these, as with most of the changes to our requirements, being the consideration of the default program. As well, more language was trimmed down throughout the use case for readability's sake.

Another request the customer had was that we add an AI entity to the UML diagram and a little more depth to the diagnostic entity. To satisfy this request an AI entity was added with with an attribute that it handles the construction sequence. It also has relationships with the Supervisor, control interface, and construction menu. We also added more relationship lines between a few of the entities to the UML diagram, where it was determined that they do, in fact, need to have a relationship. This was discovered when the map prototype and Control interface prototype were created.

The customer requested that the team specify the list of commands under our functional requirements. The list of commands were then switched to 'controls' instead, and the controls were specified in this list. The customer also commented that saying 'list of reports' was also vague, so that was changed to 'display diagnostics, rover, satellite and assembly colony reports.' Life support was not mentioned in the requirements when it was supposed to be, so this was also added to the documentation. Another requirement was added as well. It stated that the rover and assembly colony would go into default operation whenever there was no communication between the system, rover and assembly colony due to a damaged or blocked satellite. This default operation would only go on until communication has been reestablished. This new requirement was added after the customer realized two contradicting things in Use Case 3. First, there would be no way for the supervisor to have manual control of the rovers or assembly colony if the satellite is damaged. Second, the rovers and assembly colony should really be much more independent of supervisor control. These two reasons are what drove the change to be made in Use Case 3 and the last of the requirements .

When it came to the non-functional requirements, it was requested that we do not say ‘in real time’, since that would not be possible. Therefore, ‘in real time’ was changed to ‘as quickly as possible’, which would be more realistic.

Meeting with the Customer

We had a meeting with the customer on Wednesday October 23rd the meeting lasted about an hour. We each presented the prototypes we had made. After we finished our presentation, the customer went through our requirements and use cases and discussed his concerns, as well as the revisions that he would like done. Some of the concerns he had were that the colony would have delayed communication with the supervisor due to the distance away from each other, so no real time interactions could take place. He also wanted a little more detail on some of the flow of events in the use cases and for us to address the issue that we never mentioned anything about the AI used to run the colony when it wasn’t under manual control.

Contributions from each member:

Austin Crow: Created data flow diagram, formatted report to fit one style. Made sure all language and style fit together into a cohesive document. Created login and override paper prototypes.

Justin Phillips: Wrote Use Case 2, created UML diagram, paper prototype map and control interface, added the updates that were requested by the customer to use case 2, UML diagram, added those changes to Changes Made, added Meeting with customer notes

Ben Golden: Use case 1 and message sequence diagrams for the three use cases listed. Withdrew from class shortly after HW1

Daniela Muniz: Updated use case 3, updated requirements definitions, and updated requirements specifications, created construction menu and diagnostic paper prototypes.

Max McKinney: Took over Ben’s duties: Use Case 1, sequence diagrams. Proofreading and formatting.

Paper Prototypes

Logged IN [Username] [LOG OUT]

SPACE FORCE

SELF ASSEMBLER CONTROL INTERFACE

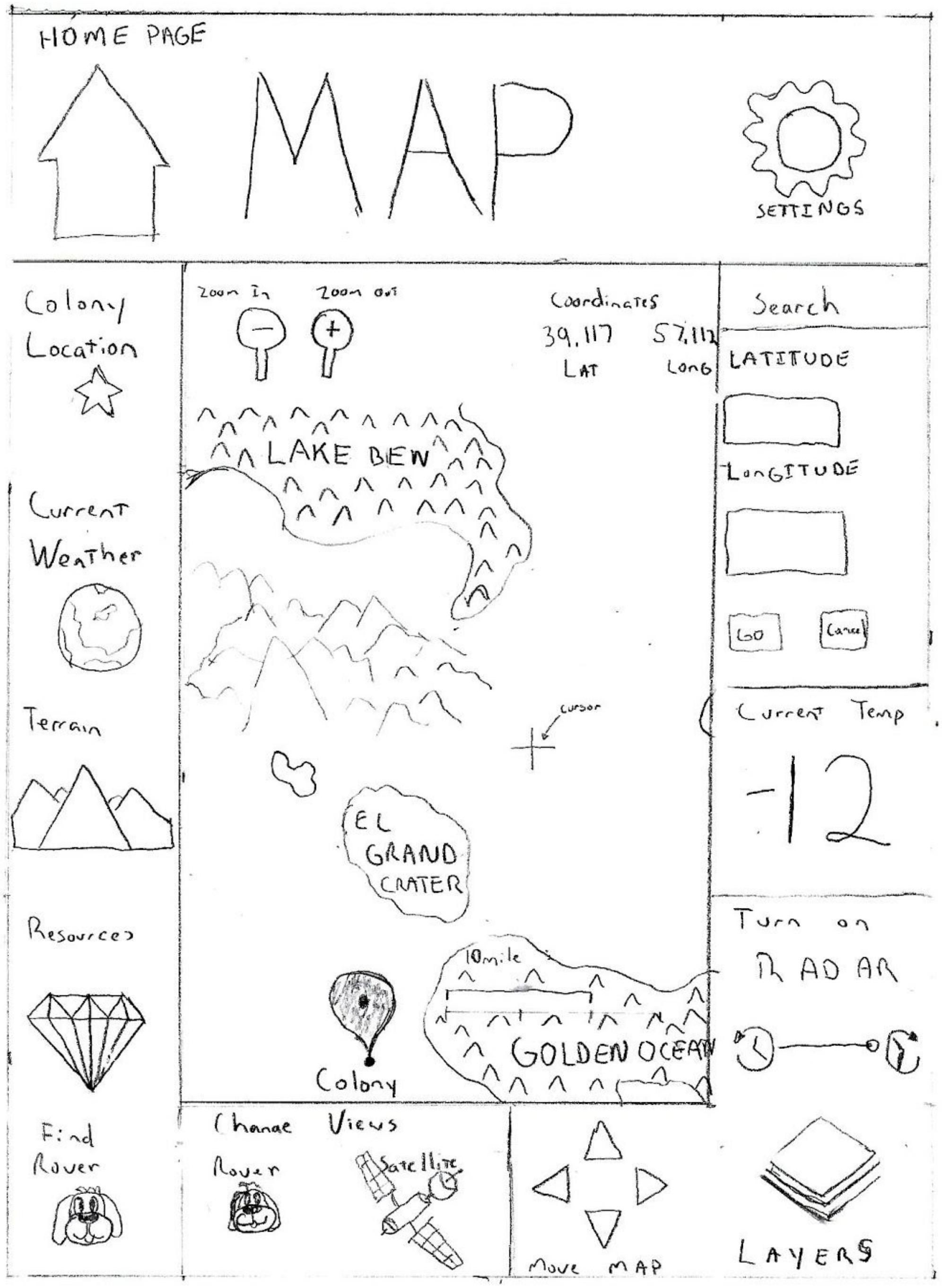
ROVER CONTROL

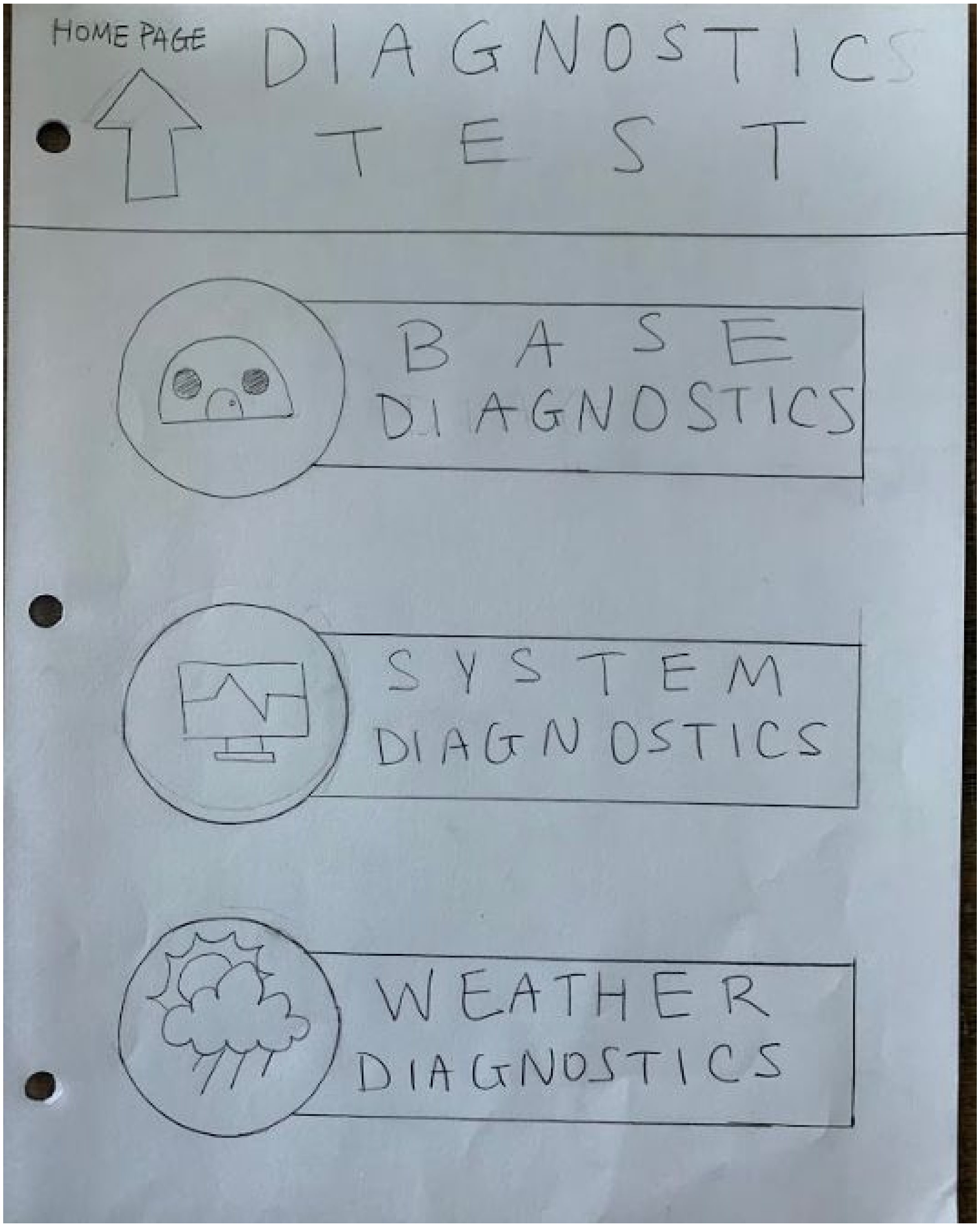
COMMUNICATION CONTROL

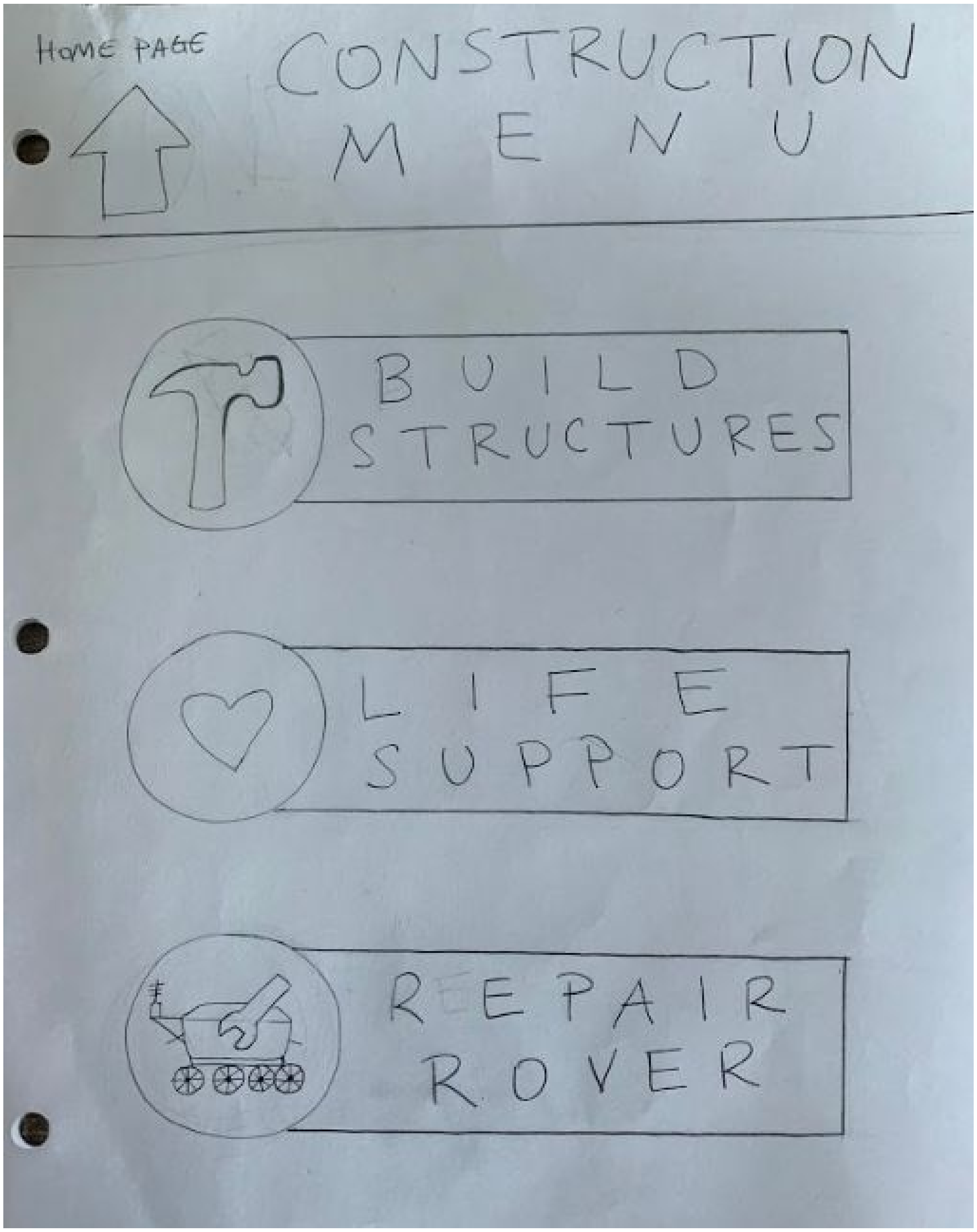
ASSEMBLER CONTROL

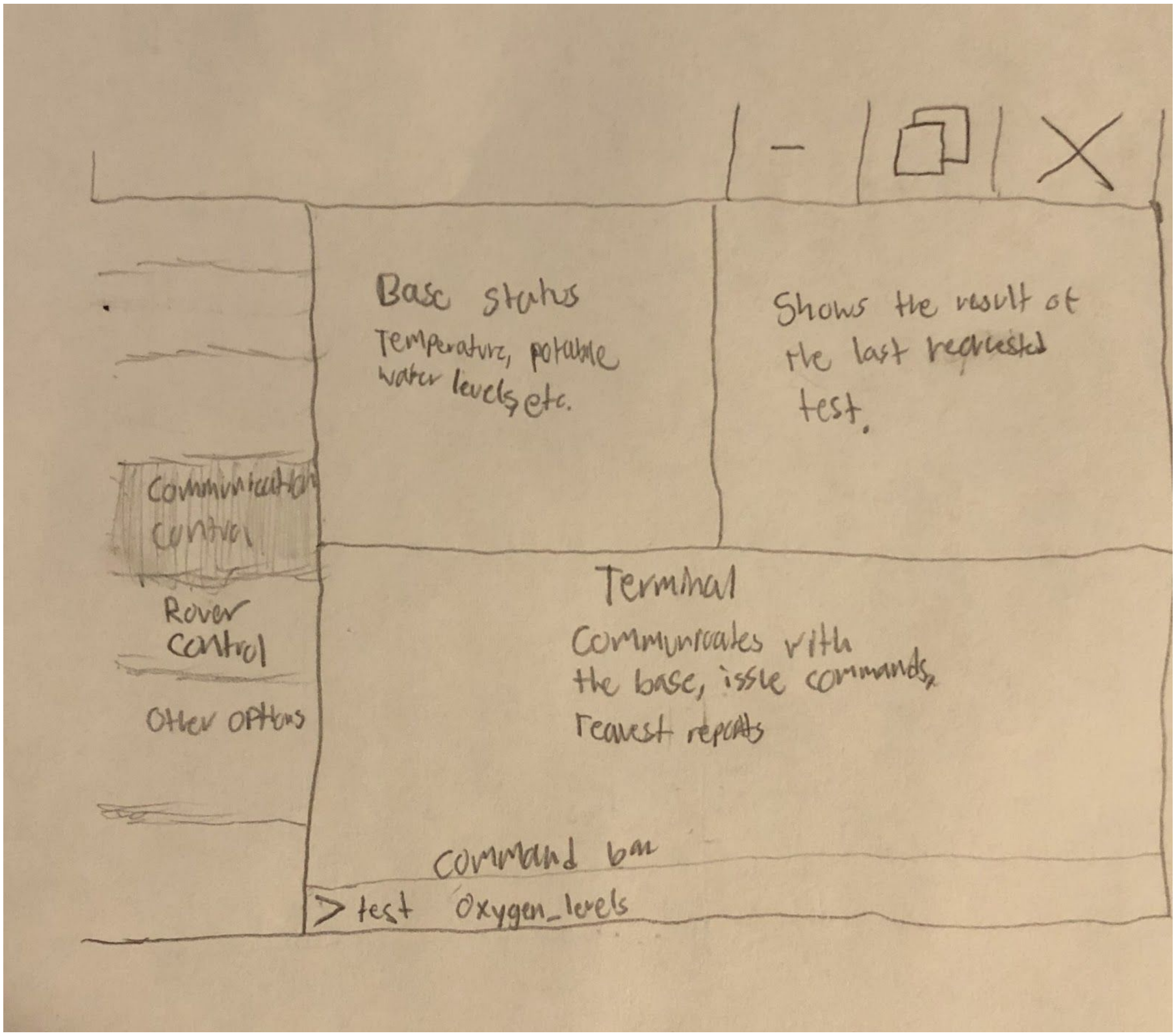
MAP	CONSTRUCTION MENU	DIAGNOSTIC TEST

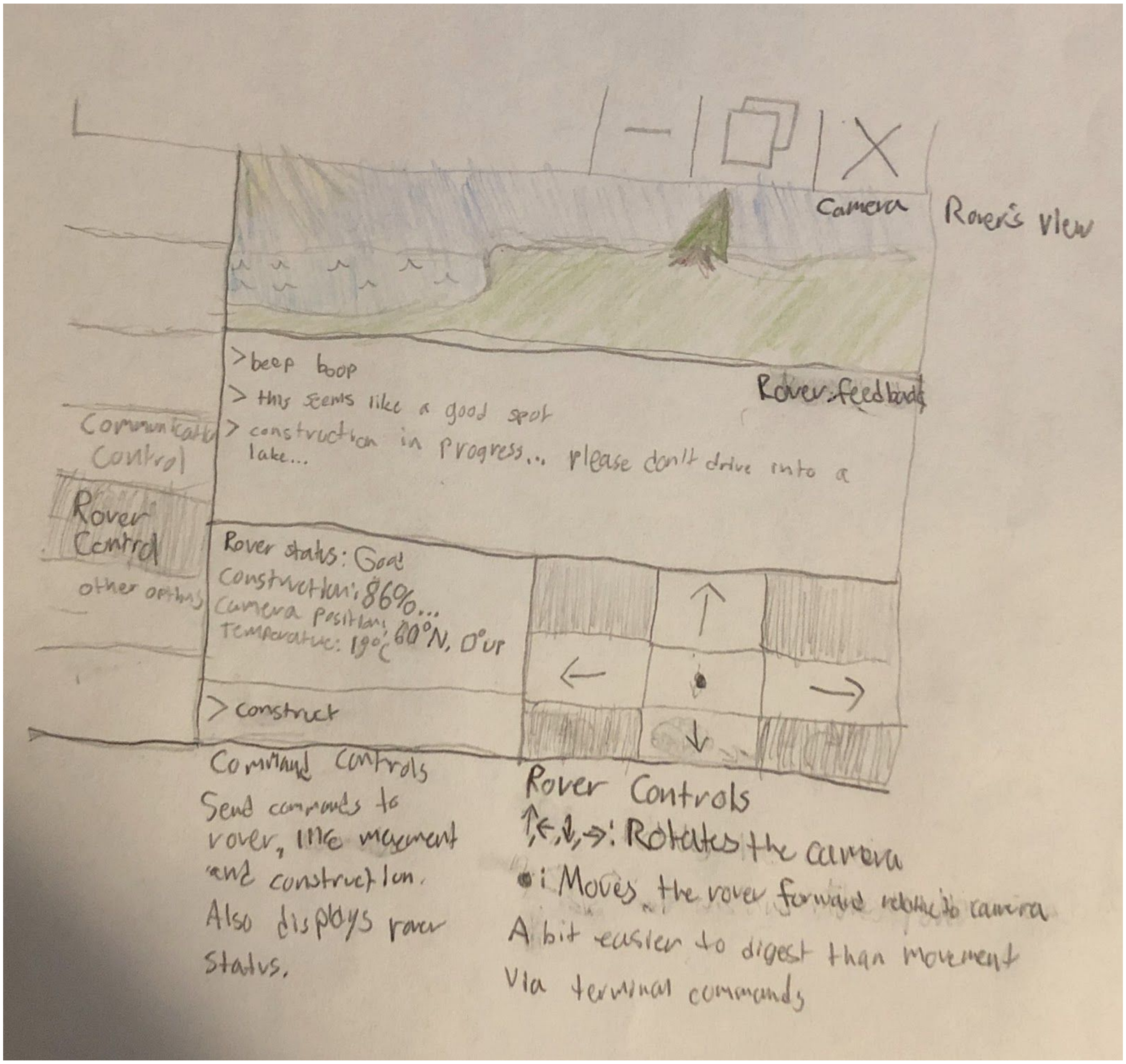
MAP

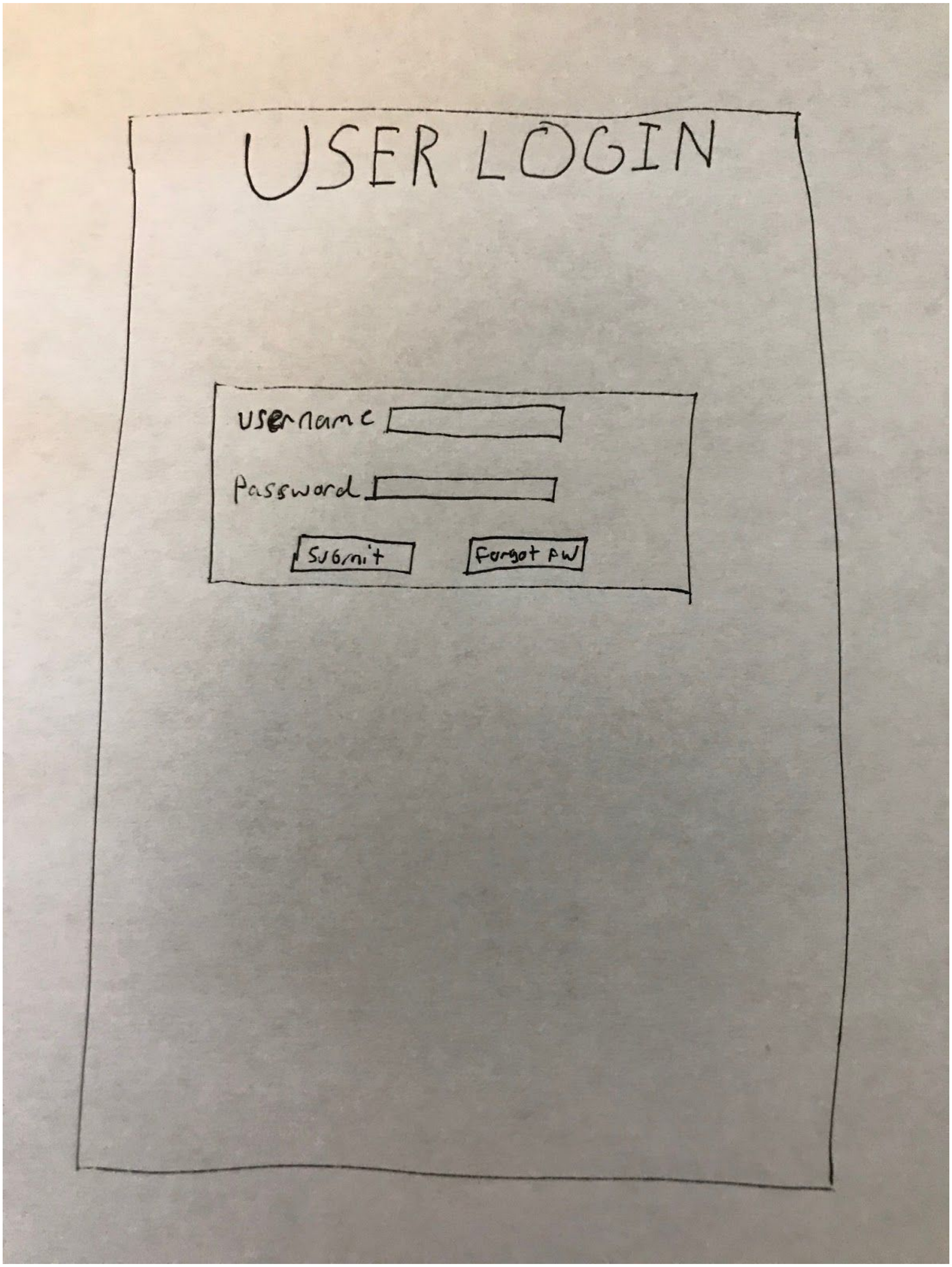

HOME PAGE
MAP
SETTINGS
Colony Location
Zoom In
Zoom out
Coordinates
39.117
57.112
LAT
LONG
Search
LATITUDE
LONGITUDE
Go
Cancel
LAKE BEW
Current Weather
cursor
Current Temp
-12
Terrain
EL GRAND CRATER
Turn on RADAR
Resources
10mile
GOLDEN OCEAN
Colony
Find Rover
Change Views
Rover
Satellite
Move MAP
LAYERS

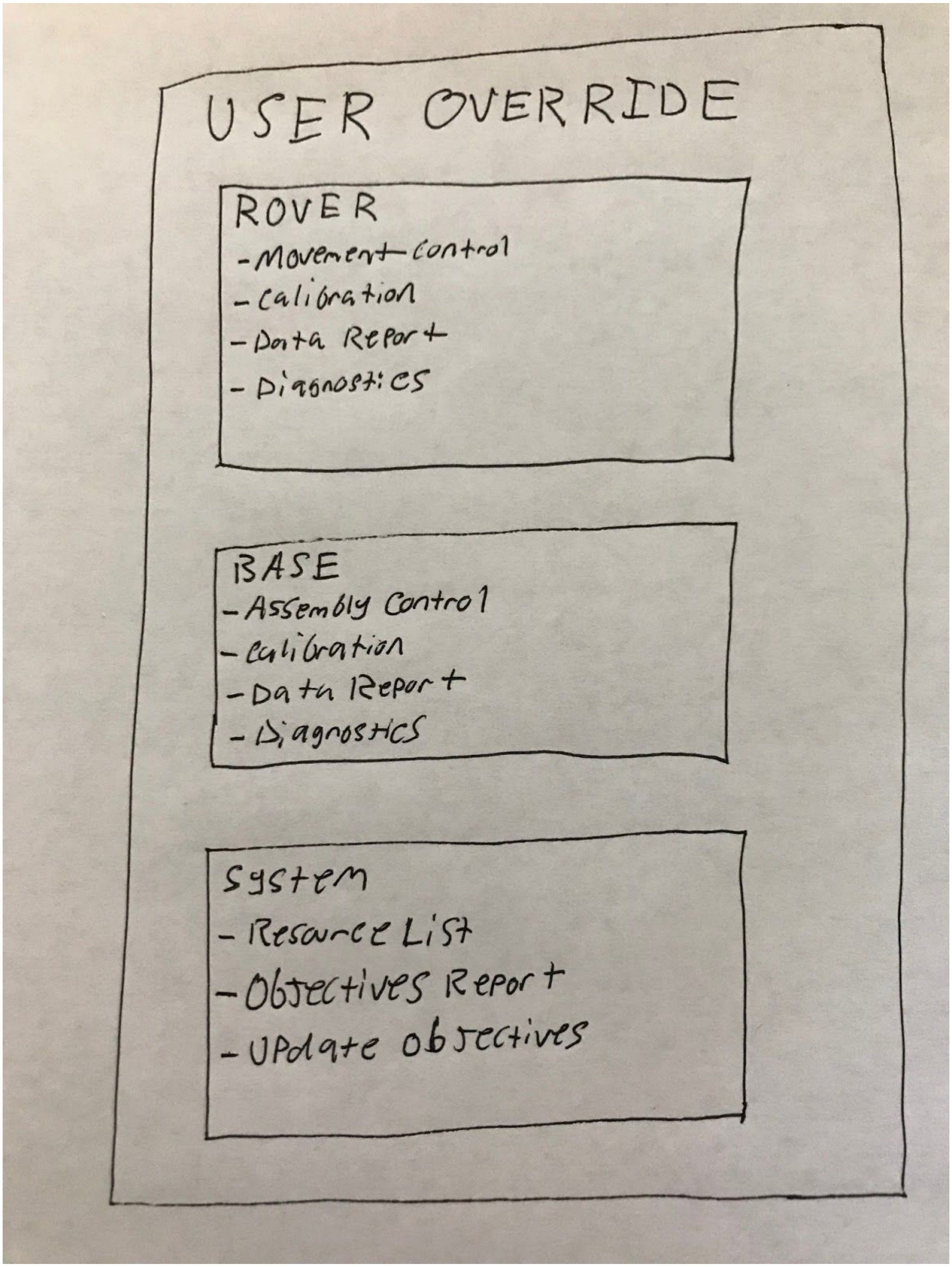

HOME PAGE
DIAGNOSTICS TEST
BASE DIAGNOSTICS
SYSTEM DIAGNOSTICS
WEATHER DIAGNOSTICS


HOME PAGE
CONSTRUCTION MENU
BUILD STRUCTURES
LIFE SUPPORT
REPAIR ROVER


Base status
Temperature, potable water levels, etc.
Shows the result of the last requested test.
Communication control
Rover control
Other options
Terminal
Communicates with the base, issue commands, request reports
Command bar
> test oxygen_levels


Camera
Rover's view
>beep boop
> this seems like a good spot
> construction in progress... please don't drive into a lake...
Rover feedback
Communication Control
Rover Control
other options
Rover status: Good
Construction: 86%...
Camera position: 60°N, 0°ur
Temperature: 19°C
> construct
Command Controls
Send commands to rover, like movement and construction. Also displays rover status.
Rover Controls
↑,←,↓,→: Rotates the camera
●: Moves the rover forward relative to camera
A bit easier to digest than movement via terminal commands


USER LOGIN
username
password
submit
Forgot PW


USER OVERRIDE
ROVER
-Movement control
-calibration
-Data Report
-Diagnostics
BASE
-Assembly Control
-calibration
-Data Report
-Diagnostics
System
-Resource List
-Objectives Report
-Update objectives