Software Requirements Specification

for

Antarctica Glimpse Network

Prepared by Kelvin David 1345360

Waikato University

09/03/2020

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is outlining the technical requirements and realization of the *Antarctica Glimpse Network* product requested by the NZASA. It will explain the technical features of the product, it's interfaces and the risks it may face. The design decisions in this document will be based upon the Project Initiation Document and any additional information elicited from the client.

1.2 Intended Audiences

- *Client/Project Manager:* Who want the specification to help dictate their decisions for future development, make them aware of the possible risks and if the features facilitate their desires.

- *Developers of the Product:* Who want to know all the technical requirements, constraints and risks, so they know what conditions to develop around

1.3 Product Scope

The product is a software system that broadcasts livestreams from rover cameras in Antarctica and enables NZASA portal administrators to manage rovers and their cameras. It also allows any registered user to view the rover livestreams of their choice. Registered user may upgrade to premium and control cameras associated with their account.

2. General Description

2.1 Defined User Roles

Registered Member

Registered user is the most common user type to use the product. They will mainly use the product to stream the content produced by their selected rover. Their characteristics include:

- They can search and select available rovers both on a list and a map.
- They can select a camera on a rover and stream its content.
- They will be able to upgrade to Silver/Gold status after making a yearly subscription payment.

Registered Silver Premium Member

Silver members is the first tier of premium member, they have all the same permissions as a registered member with added benefits. Users in this role want to be able to view and control rover cameras. Their characteristics include:

- They will need to pay a yearly subscription to keep this status.
- They will have two cameras of two different rovers associated with their account.
- They will be able to control the cameras associated with them.
- They can upgrade to Gold status after making a larger yearly subscription payment.

Registered Gold Premium Member

Gold members is the second tier of premium member, they get all the same permissions as a registered member with added benefits. Users in this role want to be able to view and control rover cameras and change their camera associations. Their characteristics include:

- They will need to pay a yearly subscription to keep this status.
- They will have three cameras of three different rovers associated with their account.
- They will be able to control the cameras associated with them.
- They can change one of their rover associations once a month.

NZASA Portal Administrators

NZASA portal administrators have unrestricted permissions over managing their rovers and cameras. Users in this role want to be able to manage all rovers, cameras and user accounts. Their characteristics include:

- They will have access to all registered user accounts.
- They will have control of all cameras associated with the user accounts.
- They will be able to manage the rovers associated with the user accounts.

3. Requirements

3.1 Functional Requirements

User Signup/Sign in: All users should be able to register an account and login to that created account.

Rover Selection/Stream: All registered users should have the ability to do the following with any rover camera connect, disconnect and view the given stream.

Rover Camera Control: Gold/Silver should have the additional function of controlling any camera associated with them.

Rover Search: All registered users will be able to search for available rovers both on a list interface and map interface. Gold/Silver should be given direct access to their associated rovers using a separate interface available to them without the need of searching.

Upgrade Account: All non-premium registered members can upgrade their user status to premium (gold/silver) by making an online payment for a yearly membership at any time. This should also be the same for silver members upgrading to gold.

Administrator Control over Accounts: Any administrator should have the permission to manage any registered user account, the ability to manage any rover and camera associations with the account.

Administrator Control over Rovers: Any administrator should have the permission to control any camera and rover.

3.2 Functional Requirements Analysis & Classifications

Rover Selection/Stream

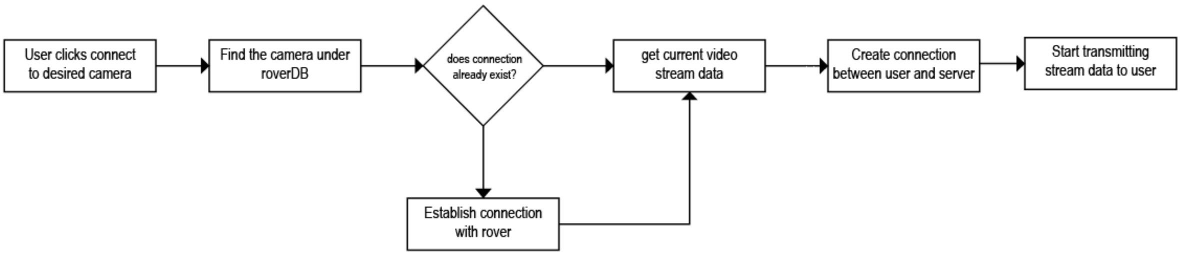
Priority: High

Classification: This is a qualitative and emergent requirement as it spans across the frontend, cloud-server and rovers. It also describes general properties.

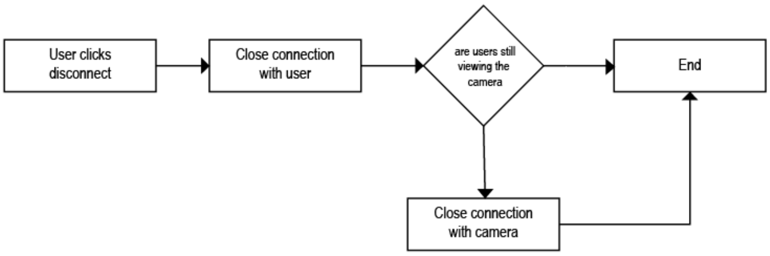
When the user selects a camera stream of a rover a connection request for that camera will be sent to the cloud server. The server will then find that camera in its database, if a connection is already established it will retrieve the streamed data and send back towards the user, otherwise it will establish the connection first. A connection will be created between the user and the server with a continuous stream of video stream data.

When users disconnect from a camera it will send a close request to the server in which the server will cut its connection with that user. The server will check if there are no other users requesting video stream from the camera that was being viewed by the recently disconnected user, if there are none then server must close its connection with that camera as well.

User Connection Process


User clicks connect to desired camera
Find the camera under roverDB
does connection already exist?
get current video stream data
Create connection between user and server
Start transmitting stream data to user
Establish connection with rover

User Disconnection Process


User clicks disconnect
Close connection with user
are users still viewing the camera
End
Close connection with camera

Rover Camera Control

Priority: Medium

Classification: This is a qualitative and emergent requirement as it spans across the frontend, cloud-server and rovers. It also describes general properties.

The premium user will have a click-based interface with their video feed that allows them to move the camera left and right. When either control is clicked it will send a movement command request to the cloud server. The server will receive the request and check what rover camera that user is connected to, then will send the command to the rover camera.

Administrator control over Accounts/Rovers

Priority: Medium

Classification: This is a component-specific, qualitative requirement as it is specific to database control and is a general property.

When administrators send SQL query requests to the cloud system (which have higher priority than user requests) to make changes to the user database and rover database. The range of commands will allow them to change/reset username and passwords, manage the account's rover associations and delete accounts from the database.

When an Administrator makes a query, their request is pushed onto the top of the queue and will execute after the currently running system event. They can also send requests to access the rovers for maintenance and relocation if needed.

User Signup/Sign in

Priority: Medium

Classification: This is an emergent, qualitative requirement as it involves communication from frontend to backend and is a general property.

Users will be asked to enter a username and password in the given input boxes. Users can press register and the username/password will be sent to the cloud server to see if there are any matching usernames, if not it will save it and refresh the page to the home page.

Users can press login and the username/password will be sent out to the cloud server to see if there are any matching usernames/passwords, if not an error message must appear saying 'incorrect username/password'. If there is a match it will refresh the page to the home page.

Rover Search

Priority: Low

Classification: This is an emergent, qualitative requirement as it involves communication from backend to frontend and is a general property.

The server should be storing the information about each rover's coordinates, ID and cameras under the rover database and should be sending an updated list of the rovers and their current coordinates to the frontend users every 1hour to be displayed on a list/map interface. When a rover camera is flagged with (inService=false) the rover name must appear red and non-interactive.

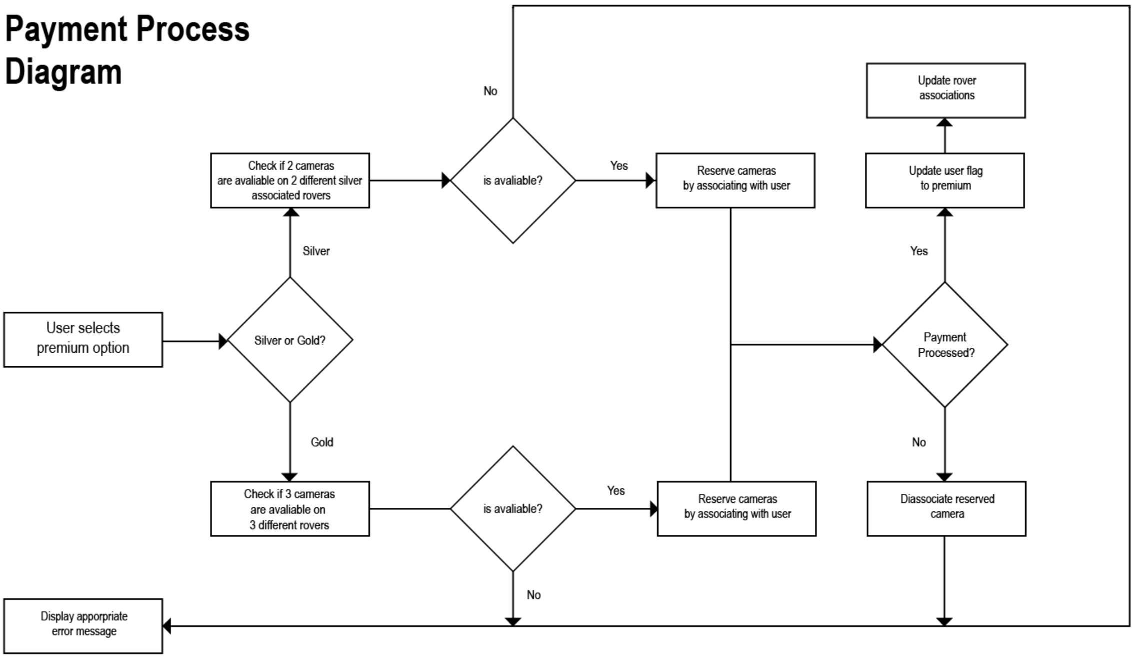
Upgrade Account

Priority: Low

Classification: This is an emergent, qualitative requirement as it involves communication from frontend to backend and is a general property.

The options for premium account are Silver and Gold status. Gold will be slightly more expensive than silver. When a registered user upgrades to premium they will go through a payment process. The sever will receive a subscription request and check for available rovers if there a no rover cameras available then it will refuse the pay otherwise it will go through. Once the payment has been successfully processed the server will flag their account as premium (which ever tier they pay for) and update their account with the extra permissions and assign them there associated rovers (3 cameras on 3 different rovers for gold; 2 cameras on 2 different rovers for silver).

It is suggested to implement PayPal to process the payments. This is best option for the moment as the product's subscription is a one-time payment. It can automatically handle failed payments on its end and doesn't require users to make an account to pay with debit/credit cards, it can also display currency conversions if the user is a different country and payments are sent straight to the client's PayPal rather than sending the credit numbers to the cloud's database, remedy that security issue.


Payment Process Diagram
User selects premium option
Silver or Gold?
Silver
Check if 2 cameras are avaliable on 2 different silver associated rovers
is avaliable?
No
Yes
Reserve cameras by associating with user
Gold
Check if 3 cameras are avaliable on 3 different rovers
is avaliable?
Yes
Reserve cameras by associating with user
No
Payment Processed?
Yes
Update user flag to premium
Update rover associations
No
Diassociate reserved camera
Display apporpriate error message

3.3 Nonfunctional Requirements

Web/Mobile frontend: It's expected to have a web/mobile frontend that human users will interact with the system. Through this front end they should be presented a human friendly interface that allows to connect, stream, control, and disconnect from the rovers with ease. It is also expected to be able to communicate to the rovers through the backend.

Cloud-based backend: The cloud-based backend should responsible for store the state of the system meaning the information of all users, the permissions they hold, the location of the rovers and their associations. It should also handle the information transfer between the various connected frontends and the rovers they are currently communicating with.

Cameras can have one association at a time: Its expected that all rover cameras should only be associated with one premium user at a time.

Camera user role association assignment bias: It's expected that the assignment of rover association to premium users is done randomly, but when assigning a silver association, the system must only assign it to rovers with other silver associations.

Livestream quality/User Scalability: The system must be designed so that the system can scale when needed, due to varying level of user load. This means that it must be able to adapt to the current user load and change respectively.

Premium User Limit Regulator: The system must be able to handle when the max number of premium users is hit and limit users to only one year of subscription time

Information/data secure transfer: The system must use a secure transportation protocol went transfer data between the frontend and the backend.

3.4 Non-Functional Requirements Analysis & Classifications

Web/Mobile frontend

Priority: Medium

Classification: This is an emergent, qualitative requirement as it involves communication from frontend to backend and is a general property.

The following languages are to be used to develop the web frontend:

- HTML
- CSS
- jQuery/Javascript
- PHP

It is also expected to use the AJAX web development technique to update webpage changes as it is slightly faster to use a combination of AJAX + PHP than just original PHP as it updates individual parts of the site rather than the whole page. This method is compatible with the updated popular browsers like Chrome/Internet Explorer, Firefox and Safari, they are also compatible with the mobile counterparts, though browsers lower than Netscape v7.1 will not be compatible.

The mobile frontend will be implemented as a browser-based application using these following languages:

- Java (Android)
- Swift/Objective-C (IOS)

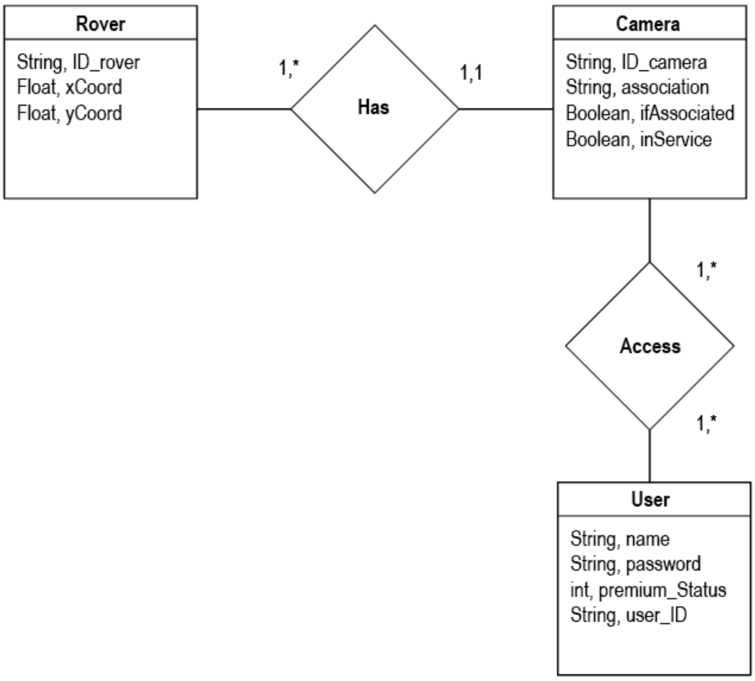
The mobile front end will just be the same implementation of the web frontend, but the interface will be scaled and modified to be more touch-based friendly. It is expected that both frontends can communicate with the backend cloud-base using HTTPS requests.

Cloud-based backend/User Scalability

Priority: High

Classification: This is an emergent, qualitative requirement as it involves communication between frontend, backend and rovers and is a general property.

The cloud-based backend must store two databases, a user database and a rover database and uses SQL to manage the data held in both databases. Structure as shown:

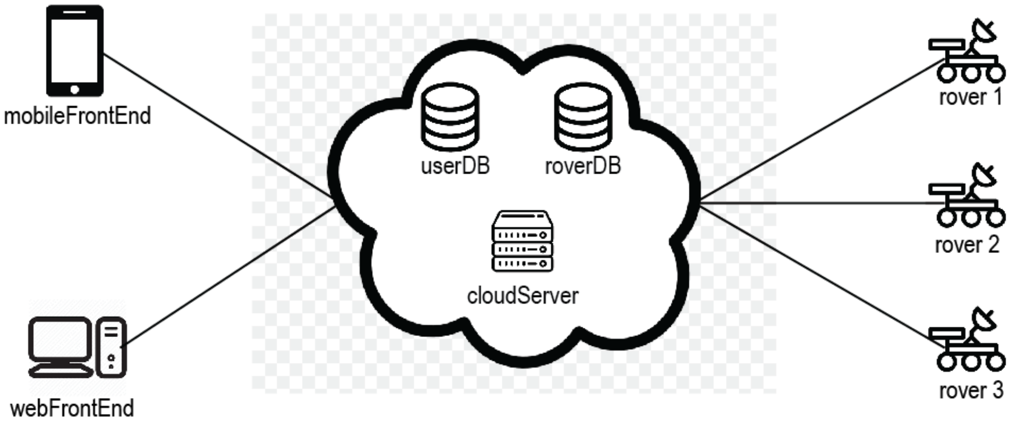


It must support media transfer to handle the transmission of the video stream data received from the rovers and the video requests from the frontend users. For this implementation I believe Amazon AWS is the best cloud-based server to use as it is completely customizable in terms of features and is cost effective with the 'per second billing' where it will charge only what you use so you won't need to worry about maximizing usage. It also supports MYSQL databases (Amazon RDS) and queues (Simple Queue Service).

Using Amazon AWS is highly scalable as you can continue to add more features to the server once you want to expand your products features. It is also cost effective in terms of scalability as the less of the system you use the price is cheaper.

The cloud server must handle the communication between the rovers, users/admins and databases by HTTPS requests. Requests must be handled using a first-in-first-out queue to work through each request, administrator requests are immediately pushed to the front of the queue.

The diagram below demonstrates how the cloud is implemented into the system:



Cameras can have one association at a time:

Priority: High

Classification: This is a component-specific, qualitative requirement as it only involves the rover database and is a general property.

The cloud system holds a rover database in which each camera holds a Boolean of whether they are associated with a user or not. When the system selects a rover to assign it will send a SQL query to the database for a random rover camera with 'ifAssociated=false' flag. Once retrieved it will input the user under 'association' and set the 'ifAssociated' flag to true. When a camera needs to be dissociated it will execute the opposite 'ifAssociated=false' and 'Associaton=null'.

Camera user role association assignment bias

Priority: Medium

Classification: This is a component-specific, quantitative requirement as it only involves the rover/user database and is a has metrics involved in the timeout.

When the system is assigning a silver user the system must check if the randomized camera belongs to a rover with other silver associated cameras, if it is not it will need to re-select.

There is a risk with this method that when searching for a silver association the system may be caught in an infinite loop due to having not enough silver associated rover cameras. To remedy this the system will attempt this for 10 tries until it times out, because silver users cannot change cameras this situation should only occur during a new subscription, so if it times out it will treat is as a maxed out silver subscriptions base and reject the subscription.

Livestream quality

Priority: Low

Classification: This is a component-specific, quantitative requirement as it involves the video output/transfer and has metrics involved in latency.

The livestreams must be encoded with H.265 using the HLS protocol for streaming. HLS is a protocol designed for low latency streaming promising 10s/below latency and can run-on low-grade technology. Though HLS is only natively supported on mobile/safari so running it on other browsers requires a HTML5 embedded video player, browsers without HTML5 cannot be supported (all popular browsers are supported).

Premium User Limit Regulator

Priority: Low

Classification: This is an emergent, quantitative requirement as it involves communication from backend to frontend and has metrics involved in camera checks.

When a new a subscription request is sent the server must check if there are enough cameras to accommodate the user. If the new request is for silver status it must need 2 cameras (on silver associated rovers) to be available and for gold status it must need 3 cameras (any association) to be available. If these requirements are not met, then the system must reject the subscription request and display an error message 'Sorry not enough rovers, try again later'

To prevent the risk of having subscriptions always maxed out. Subscriptions must only last for a fixed one year from when the payment has gone through, after which the cloud server should update the user's account premium status to 0 (common) which will remove their associations and permissions.

Information/data secure transfer

Priority: High

Classification: This is an emergent, quantitative requirement as it involves communication from frontend to backend and has metrics involved in load time.

The system must use HTTPS requests/responses when communicating between the cloud Webserver and the mobile/web frontend users. So, when sending the chunks of video stream to the viewers, when AJAX requests for a page refresh using plain text or when user information is sent from the frontend to the server it will use HTTPS to encrypt the data during transfer. The draw backs of HTTPS are that it is 0.47s slower load time average compared to HTTP and administrators must re-generate keys and certificate when the SSL certificate expires.

4. Conclusion

4.1 Conclusion

The *Antarctica Glimpse Network* will provide the enthusiasts and researchers access to valuable data samples of Antarctica's surface and will continue to develop to make it more accessible to everyone. This documentation realizes the first iteration of the product will be classified as a Minimum Viable Product (MVP) as it will provide only the smallest and most vital of features for the customers' need. This way it will be easier and cheaper to produce in comparison to a full-fledged product. Through customer feedback developments and alterations can be made to the future iterations.