# Group QI QUADCOPTER Design Document

Version 0.1
05 MAR 2014

**Supervisor:**
Einar Sundgren
mail@einarsundgren.se

**Scrum Master:**
Gabriele Kasparaviciute
gabekasparas@gmail.com

**Members:**
James Eagle
Jani Pasanen
Henrik Persson
Aurélien Franky
MaiPhuong Van
Gabriele Kasparaviciute

Gothenburg University

# Table of Contents

## Introduction

The Software Design Document is a document to provide documentation, which will be used to aid in software development, by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information.

## Purpose

The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to built. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

## Scope

This Software Design Document is for a base level system which will work as a proof of concept for the use of building a system that provides a base level of functionality to show feasibility for large-scale production use. This Software Design is focused on the base level system and critical parts of the system. For this particular Software Design Document, the focus is placed on generation of the documents and modification of the documents. The system will be used in conjunction with other pre-existing systems and will consist largely of a document interaction facade that abstracts document interactions and handling of the document objects.

## Definitions, Acronyms, and Abbreviations

- **Raspberry Pi** - The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games. It also plays high-definition video. We want to see it being used by kids all over the world to learn programming.
- **Yaw –** (turning left and right) is controlled by turning up the speed of the regular rotating motors and taking away power from the counter rotating;
- **Roll** (tilting left and right) is controlled by increasing speed on one motor and lowering on the opposite one.
- **Pitch** (moving up and down, similar to nodding) is controlled the same way as roll, but using the second set of motors.

# Architecture Design

## Overall design

The quadcopter is assembled from scratch, the components have been bought separately and we will assemble it ourselves. To pilot the quadcopter an advanced autopilot called Ardupilot will be used. It's based on the Arduino Mega and runs open source software. Commands will be sent to the autopilot from a laptop over WiFi. The Raspberry Pi will be mounted on the quadcopter and transmit these commands to the Ardupilot. A program will be used to stream the video from the Raspberry Pi. Furthermore,  XBOX 360 controller will manage control of the aircraft.

## System Hardware Architecture

Frame

- Arms
- Center plates
- Machine screws & bolts

Power System

- battery: LiPo 3S 4000mAh
- 4 x ESC (motor controller)
- 4 x  motors

Radio System (remote controller)

## System Software Architecture

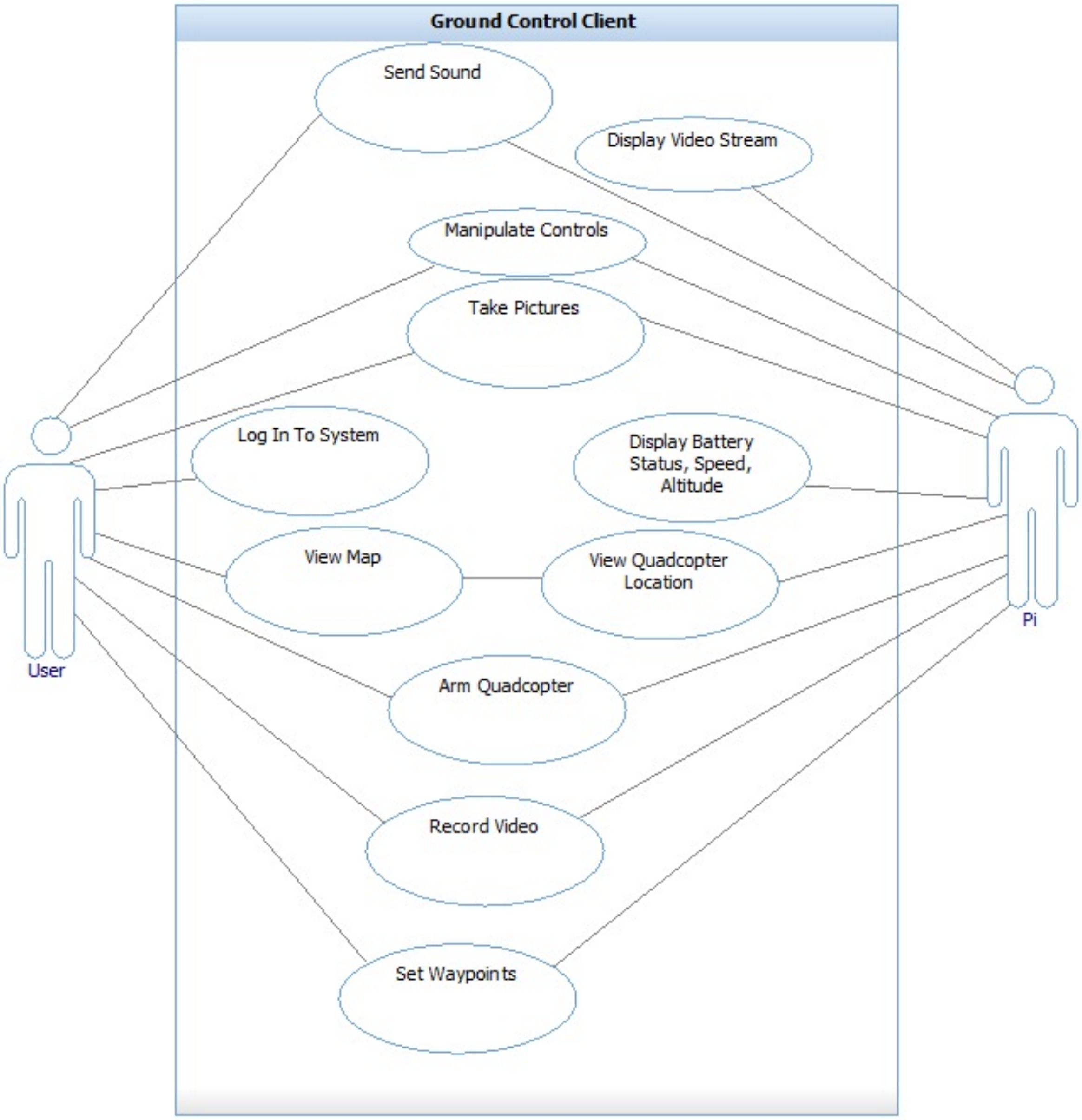
# USER STORIES

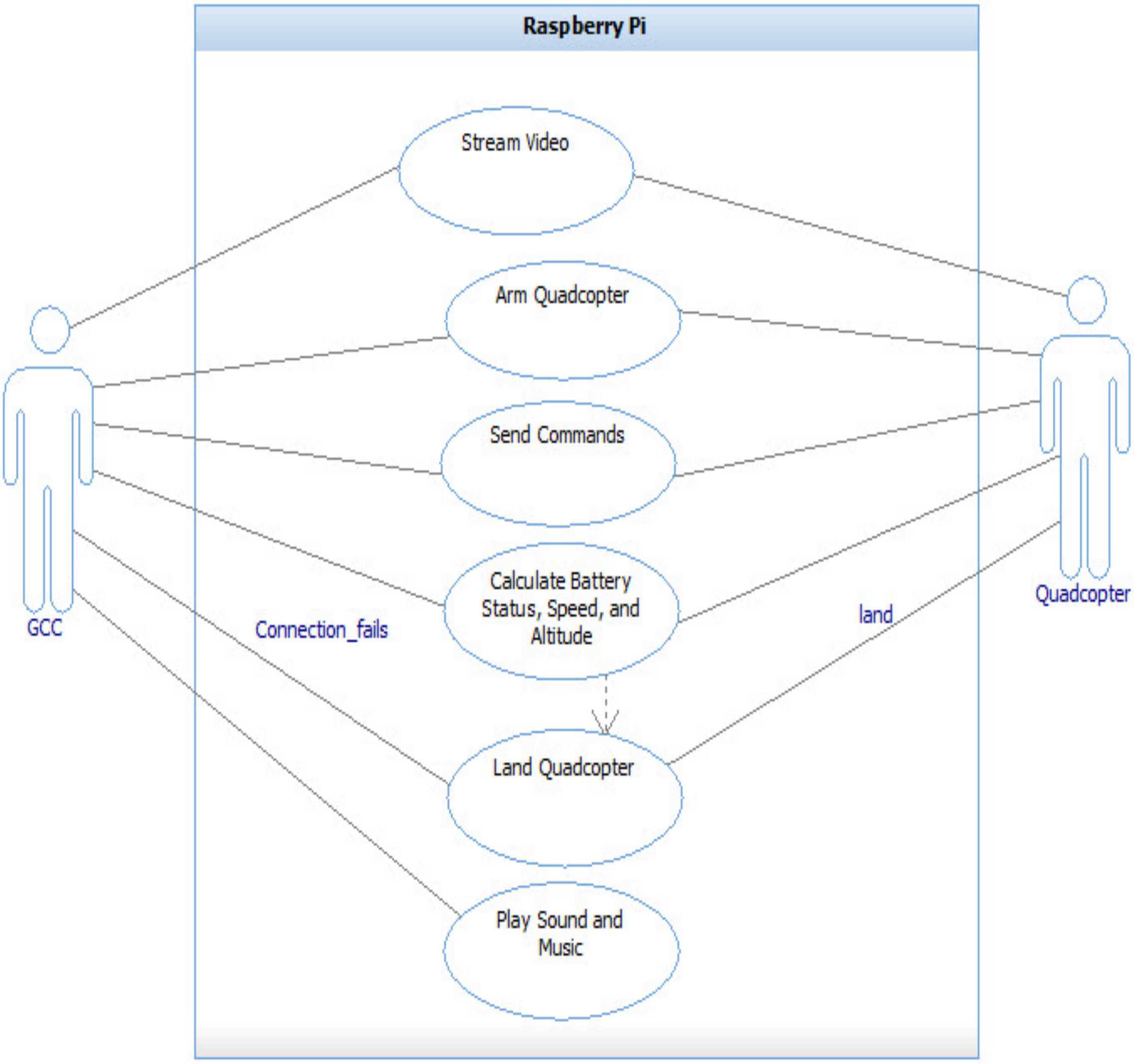
- As a user, I would like to be able to control the quadcopter regardless of distance.
- As a user, I would like to control my quadcopter from my computer/mobile device through either a keyboard or controller or touch interface.
- As a user, I want the system to react to my commands instantly.
- As a user, I would like the quadcopter to be able to do sweet flips.
- As a user, I would like it to be able to stream real-time video in order to fly the quadcopter outside of my field of view.
- As a user, I would like to see speed and altitude displayed on screen
- As a user, I would like to record a video/photo.
- As a user, I would like to be notified if the battery is critically low, to avoid accidents.
- As a user, I would like my quadcopter to land safely in case of emergency.
- As a user, I would like to be the only one who is able to connect and control the quadcopter at any given moment.
- As a user, I would like my quadcopter to play music/sound
- As a user, I would like to view my quads position on a map
- As a user, I would like to set waypoints for my quadcopter to fly to
- As a user, I would like my quadcopter to operate safely

## Additional user stories

- *(As a user, I would like my quadcopter to react to motion and its environment, in order to steer it without controller.)*
- *(As a user, I would like my quadcopter to be able to transport stuff, for example, sushi and beer.)*
- *(As a user, I would like to use a microphone to send sounds to pi)*
- *(As a user, I would like my quadcopter to map out corridors)*
- *(As a user, I would like to script a sequence of commands for my quad)*
- *(As a user, I want to be guided through the software)*

## USE CASE DIAGRAMS




Raspberry Pi
Stream Video
Arm Quadcopter
Send Commands
Calculate Battery Status, Speed, and Altitude
Land Quadcopter
Play Sound and Music
GCC
Connection_fails
land
Quadcopter

# USE CASE DESCRIPTIONS

## 1. The user arms the quadcopter

**Description**
In order to fly the quadcopter, the user needs to arm it. Having to arm the quadcopter before it can fly minimizes the risk of injuries.

**Trigger**
The user wants to fly the quadcopter.

**Actors**

1. user/pilot
2. ground control software
3. raspberry pi/quadcopter

**Preconditions**

1. quadcopter has battery and is charged
2. computer/mobile phone has an internet connection
3. computer/mobile phone has enough battery

**Goals**
fly the quadcopter

**Failed conclusion**
the user was unable to arm the quadcopter

**Steps of execution**

1. log in to ground control software
2. the software runs pre-flight checks
3. user is notified how the preflight checks went
4. the user presses the “ARM” button
5. the user is notified that the quadcopter is now armed and ready to fly
6. the propellers start turning
7. to disarm the quad the user throttles down and waits 5 sec

## 2. The user records a video

**Description**
This is what the user has to do in order to record a video on the SD card of the raspberry pi.

**Trigger**

1. the ground control software is on
2. the quadcopter is on

**Actors**

1. user/pilot
2. ground control software
3. raspberry pi/quadcopter

**Preconditions**

1. quadcopter has battery and is charged
2. computer/mobile phone has an internet connection
3. computer/mobile phone has enough battery
4. the raspberry pi camera is functional

**Goals**

record a video in good quality

**Failed conclusion**

1. the user was unable to record a video

**Steps of execution**

1. Log in to ground control software
2. the software runs pre flight checks
3. the user is notified that the pre flight checks were successful
4. the user sees a video stream from the raspberry pi on his monitor
5. the user presses the button “record video”
6. the user is notified that a video is being recorded
7. the user presses the button “stop recording”

***Additional***
to view the video, the user removes the SD card from the raspberry pi and plugs it in a computer.

## 3. The user views the quad location

**Description**
The user executes the following steps in order to view the quad location

**Trigger**

1. the ground control software is started
2. the quadcopter is started
3. the software went through the pre flight checks and the user is logged in

**Actors**

1. user/pilot
2. ground control software
3. raspberry pi/quadcopter

**Preconditions**

1. quadcopter has battery and is charged
2. computer/mobile phone has an internet connection
3. computer/mobile phone has enough battery
4. the quadcopters GPS module has connection

**Goals**

1. view the quad location on a map

**Failed conclusion**

1. the quadcopters location is not displayed on the map

**Steps of execution**
*if the user is on a desktop computer*

1. a small map is displayed on the top right side of the screen, the map is centered on the quadcopters location
2. to have a better view of the quadcopters location, the user can press the small map to maximize it. If an XBox controller is connected the user presses “A”
3. to close the map again, the user clicks the minimize button or presses “A”

*if the user is connected over a mobile phone*

1. a map icon is displayed on the top right side of the screen
2. to view the map, the user clicks the icon
3. the map is now full screen, the controls are not visible anymore. The quadcopter goes in hover mode
4. to close the map, the user clicks the command icon on the top right side

## 4. The user sets waypoints for the quad to fly to

**Description**
The user selects points on the map, the quadcopter then flies to these locations one after the other.

**Trigger**

1. the ground control software is started
2. the quadcopter is started
3. the software went through the pre flight checks and the user is logged in

**Actors**

1. user/pilot
2. ground control software
3. raspberry pi/quadcopter

**Preconditions**

1. quadcopter has battery and is charged
2. computer has an internet connection
3. computer is power on
4. the quadcopters GPS module has connection
5. the user is using a desktop computer

**Goals**

select locations on the map to create a route for the quadcopter to follow

**Failed conclusion**

1. the quadcopters doesn’t follow the route
2. the quadcopter skips a waypoint
3. the quadcopter crashes

**Steps of execution**

1. the user opens the map, the quad is now in hover mode
2. the user right clicks on a location
3. the user selects add waypoint from the menu
4. repeat this task for all wanted waypoints
5. to delete a waypoint select the waypoint and press delete
6. to move a waypoint, click it and drag and drop to the desired location
7. to make the quadcopter follow the waypoints, the user clicks the follow way points button. The quadcopter now flies to the first waypoint.

## 5. The user logs in to the system

**Description**
The user logs in to the system using username and password

**Trigger**

1. the ground control software is started

**Actors**

1. user
2. ground control software

**Preconditions**

1. computer/mobile phone has an internet connection
2. computer/mobile phone has enough battery

**Goals**

1. Log in to the system to fly the quadcopter

**Failed conclusion**

1. Someone is already logged in with the same credentials.
2. Wrong username or password.

**Steps of execution**

1. The user clicks on the GCC's icon on their PC or their mobile. Program opens and displays a login message
2. User types in their username and password. GCC displays the user's usernames in text and hide passwords as sequence of '*' signs.
3. User click 'Log in' to start using the GCC. If someone has already logged in, GCC will display error message and sends the user back to the login message. If user password or username is wrong, GCC will also display error message and send the user back to the login message.

## 6. The user plays sound

**Description**
The user send sound command to the Pi and the Pi will play sound

**Trigger**
the user clicks the audio button

**Actors**

1. user
2. ground control software
3. Raspberry PI/ quadcopter

**Preconditions**

1. computer/mobile phone has an internet connection
2. computer/mobile phone has enough battery
3. quadcopter has battery and is charged
4. Raspberry PI is connected to some sort of speakers

**Goals**
quadcopter plays sound/music

**Failed conclusion**

1. Raspberry PI is not connected to any speaker
2. the software is unable to send the sound
3. the software is unable to play the sound

**Steps of execution**

1. User taps on a sound icon displayed on GCC interface on computer or mobile.
2. GCC displays sound options for the user to choose.
3. User chooses desire sound.
4. Command signal sent to the PI through wireless network
5. PI receives command and play sound through the connected speaker.
6. If there is no speaker, PI send message back to GCC, GCC displays error message.

## 7. The user flies the quadcopter

**Description**
The user uses the mobile app, or a controller attached to the computer to fly the quadcopter

**Trigger**

1. the user arms the quadcopter

**Actors**

1. user
2. ground control software
3. Raspberry PI/ quadcopter

**Preconditions**

1. computer/mobile phone has an internet connection
2. computer/mobile phone has enough battery
3. quadcopter has battery and is charged
4. software went through pre flight checks and quadcopter is armed

**Goals**

fly the quadcopter directly

**Failed conclusion**

1. Raspberry PI is not connected to any speaker
2. the software is unable to send the sound
3. the software is unable to play the sound

**Steps of execution**

1. User taps on a sound icon displayed on GCC interface on computer or mobile.
2. GCC displays sound options for the user to choose.
3. User chooses desire sound.
4. Command signal sent to the PI through wireless network
5. PI receives command and play sound through the connected speaker.
6. If there is no speaker, PI send message back to GCC, GCC displays error message.

# REQUIREMENTS

## Functional

- quad can be steered via a (xbox) controller, keyboard, or mobile device
    - controls for throttle, roll, pitch, etc. are mapped on the keyboard/controller/mobile app
- quad streams video from the first person view to a computer/mobile device via internet
- quad streams altitude, speed, battery life.
- quad receives commands via internet
- quad plays/streams sounds/music from another computer/SD card
- quad lands safely if the internet connection is lost or client is lost or its battery is low
- user needs to log in to use the quadcopter
- user can add waypoints to which the quadcopter can navigate
- user can record the video stream or take pictures
- ground control software has a log-in screen
- ground control software displays battery status
- ground control software displays the video stream
- ground control displays map and location of the quadcopter
- ground control software displays a visual representation of the control values

## Non Functional

- quad should be stable in flight and easy to steer
- ground control software should run on different operating systems/mobile devices
- video stream should be in real time, stable and display enough detail to navigate by.
- Software should be open source and extendable