Software Requirements Specification

**Version: Final**

**October 04, 2018, Changes: October 2018**

**Revision Chart**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| Draft | Steve McConnel | Initial draft created for distribution and review comments | 19/9/2018 |
| Preliminary | Stelios, Adamos, Andriani, Stefanos, Panayiotis | Second draft incorporating initial review comments, distributed for final review | 2/10/2018 |
| Final | Eleni | First complete draft, which is placed under change control | 4/10/2018 |
| Revision 1 | Eleni, Andriani, Stelios, Stefanos, Panayiotis | Revised draft, revised according to the change control process and maintained under change control | 5/10/2018 |
| Revision 2 | Eleni, Andriani, Stelios, Adamos, Stefanos, Panayiotis | Revised draft, revised according to the change control process and maintained under change control | 10/10/2018 |

**Contents**

**1.** **Introduction**

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

**2.** **Overall Description**

2.1 Product Perspective

*2.1.1* *System Interfaces*

*2.1.2* *User Interfaces*

*2.1.3* *Hardware Interfaces*

*2.1.4* *Software Interfaces*

*2.1.5* *Communications Interfaces*

*2.1.6* *Memory Constraints*

*2.1.7* *Site Adaptation Requirements*

2.2 User Characteristics

2.3 Constraints

2.4 Assumptions and Dependencies

**3.** **Specific Requirements**

3.1 Software Product Features

*3.2.1* *Feature 1*

3.2 Performance Requirements

3.3 Software System Attributes

*3.3.1* *Reliability*

*3.3.2* *Availability*

*3.3.3* *Security*

*3.3.4* *Maintainability*

*3.3.5* *Portability*

3.4 Logical Database Requirements

3.5 Other Requirements

**4.** **Appendices**

# Introduction

What is Hang Gliding?



It allows you to own an aircraft that doesn't require an expensive hangar or special pilot license and best of all you can haul it in your car or truck. You can launch under your own power, climbing and gliding as you please.

The equipment needed is minimal by aviation standards. The main items being the glider, also called the wing, and the harness. The rest of the equipment consists mostly of safety and communication gear.

The training needed to get yourself flying solo is also minimal. The basic techniques of hang gliding -- launching, turning, landing -- are fairly easy to learn. You can fly solo from the training hill and progress to higher flights, all in two days.

Safety is of utmost importance. Despite the relatively low risk involved with hang gliding there still remains the possibility of serious injury if a pilot doesn't properly prepare and exercise good judgment. Preparation and proper training go a long way towards a pilot's well being as well as enjoyment. Part of the preparations is knowing what weather conditions you will be flying in.

Since this sport is not usually done alone, there are organizations, clubs and groups to assist new pilots and to form parties to fly with.

Hang gliding is one of the simplest forms of human flight. A hang glider is a non-motorized, foot-launched wing. A Hang glider has a rigid frame maintaining the shape of the wing, with the pilot usually flying in a prone position. The hang glider wing is constructed of rip-stop nylon or Mylar over an aluminium frame.

Hang gliders are controlled by shifting the pilot's weight with respect to the glider. Pilots are suspended from a strap connected to the glider's frame (hence the name "hang" glider). By moving forward and backward and side to side at the end of this strap, the pilot alters the centre of gravity of the glider. This then causes the glider to pitch or roll in the direction of the pilot's motion and thus allows both speed control and turning.

With a hang glider, you can fly like a bird, soaring upwards on currents of air. Hang gliders routinely stay aloft for 3 hours or more, climb to elevations of 15,000', and go cross-country for vast distances.

## Purpose

The purpose of this document is to provide a detailed description of the requirements of the mobile application Light Gliding for use in hang gliding for pilots and event organizers. It will illustrate the system constraints, interface and interactions with external sensors, mobile sensors and the user.

## Scope

The Light Gliding is a mobile application based around external sensors gathering information on humidity, temperature and speed with the combination of GPS it will be able to track the flight route of the pilot live and show a directional path as well as headed direction. Accumulating all the information, post-flight, to show statistics per certain point on the route, based on speed, altitude, temperature and humidity.

The application is an alternative, more comfortable and easier way for pilots to track their flights and add of even more useful functions, with the goal of replacing the very expensive tools that are currently needed or just combine those tools with more features.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| User | Someone who interacts with the mobile phone application. |
| Pilot | The user who controls the hang glider. |
| Hang Gliding | Hang gliding is an air sport in which a pilot flies a light, non-motorised foot-launched heavier-than-air aircraft called a hang glider. |
| GPS | Global Positioning System. |
| GPS-Navigator | An installed software on mobile phone which could provide GPS connection and data, show locations on map and find paths from current position to defined destination. |
| Stakeholder | Any person who has interactions with the system who is not a developer. |
| Downdraft | Air moving downwards, usually within a thunderstorm cell. |
| Updraft | A column of rising air in the lower altitudes of Earth’s atmosphere. |
| MM | Android ® Marshmallow (6.0) |

## References

1. *IEEE Software Engineering Standards Committee, “IEEE Std 830-1993, IEEE Recommended Practice for Software Requirements Specifications”, December 2, 1993.*
2. *https://www.android.com/versions/marshmallow-6-0/*
3. *https://developer.android.com/reference/android/database/sqlite/package-summary*
4. *<https://en.wikipedia.org/wiki/Variometer>*
5. *<https://en.wikipedia.org/wiki/Hang_gliding>*
6. *<http://xcmag.com/news/encounters-of-the-liquid-kind/>*
7. *<http://www.aerospaceweb.org/question/atmosphere/q0046b.shtml>*
8. *<https://en.wikipedia.org/wiki/Density_of_air>*
9. *[https://en.wikipedia.org/wiki/Humidity#Measurement](https://en.wikipedia.org/wiki/Humidity%23Measurement)*
10. *<http://www.kids-fun-science.com/what-is-air-pressure.html>*
11. *[https://en.wikipedia.org/wiki/Atmospheric\_pressure#Measurement\_and\_maps](https://en.wikipedia.org/wiki/Atmospheric_pressure%23Measurement_and_maps)*
12. *<http://xcmag.com/news/getting-up-to-speed/>*
13. *<http://articles.latimes.com/2007/aug/30/local/me-maccready30>*
14. *<https://www.sciencedirect.com/topics/earth-and-planetary-sciences/cumulonimbus-clouds>*
15. *<https://www.metoffice.gov.uk/learning/clouds/cloud-spotting-guide>*
16. *<https://en.wikipedia.org/wiki/Cumulonimbus_cloud>*
17. *<https://wiki.sei.cmu.edu/confluence/display/java/Input+Validation+and+Data+Sanitization>*
18. *<https://www.howtogeek.com/254603/how-to-use-your-phone-as-a-barometer-or-altimeter/>*
19. *<https://cloud.google.com/speech-to-text/docs/>*
20. https://www.epa.gov/sunsafety/uv-index-1
21. *<https://www.epa.gov/sunsafety/calculating-uv-index-0>*

# Overall Description

## Product Perspective

This system consists of an offline mobile application for Android®. In its first release, the application will assist the user by tracking various statistics relevant to hang gliding and presenting them off-line of the flight. The application will also assist the user on-line (during) the flight.

The application will be using GPS to gather location data during the flight. The location data will be used to track the user in 2D space while in flight and construct the path.

Since a lot of data will be generated by the application, an offline database that is device specific will be used to store the data.

The initial release will gather data about temperature, humidity, speed, acceleration, location, altitude, elevation, of which, some data, like location, will be gathered from the phone device itself. Humidity, temperature, acceleration and UV radiation will be gathered by SGshell which is provided by SignalGenerix.

### System Interfaces

The product is an accessory to the common devices employed by hang-gliders, professional or not but does not communicate with any of them. The product communicates with a GPS system through the phone for 2D positional tracking, and SGshell to gather data about temperature, humidity, acceleration and UV radiation.

### User Interfaces

**Navigation Menu**

The system contains a navigation menu with the following entries which represent different screens:

**Home**

Home is the home screen (first screen) that comes up when the user opens the application and it will contain a small presentation of the last flight and its data, weather information and will provide the action to record a flight.

**Fly**  
Fly entry transitions the user to flight mode. During flight mode, the application will display data in real time. The user will have the ability to decide in the settings screen which data in which order will be displayed. In addition, they will be deciding how that day may be presented. More specifically, the data could be presented in chart where only the last couple of minutes are shown. The amount of minutes shown can be changed in the settings, furthermore, the data could be presented in digital form or an analog form.

**About**

About displays information about the specific version of the Application.

**History**

History page shows a list of previously recorded flights, along with their date, time and any custom name provided by the user. The entries allow clicking on them which transitions the user to a page specifically for the selected flight and will be referred as Flight.

**Flight**

Flight page presents all the gathered data in graphs where applicable as a function of time, such statistics are humidity, temperature, speed, altitude and altitude change.

**Statistics**

Statistics presents the average, minimum and maximum statistics for a specific data point. Such statistics are, top 2D speed, maximum altitude decrease/decrease rate, max/min temperature, directional acceleration.

**Flight Mode**  
Flight mode is an option which gives the ability to the pilot to get some additional information, like speed, temperature, altitude, elevation, direction, etc, during the flight. The information that will be displayed in this mode are based on the user’s preference as it can be customised. A default option will be available which will include the basic measurements that the pilot will need.

**Settings**

Settings screen provides various options the user can adjust to personalize their experience. For example, there is the option to select between the unit of measurements to appear on the statistics. [Flight mode] Also, the user can select which statistics and measurements can view during a flight.

### Hardware Interfaces

SGshell will be used to track humidity, temperature and acceleration. SGshell is provided by SignalGenerix and connection to the hardware device is done over Bluetooth®. Additionally, the communication layer between Bluetooth® and SGshell is provided by SignalGenerix.

### Software Interfaces

**Operating System**

The Operating System (OS) is device dependent, however, Android ® Marshmallow (6.0) which will be referred as MM, will be used as baseline due to its market share and compatibility between it, and later versions.

Android MM’s specification exists in reference [2]

**Database**

The system will be using SQLite®. SQLite is provided by the Operating System and as such, version number and specification number vary. Documentation for SQLite is provided by Google (see reference [2])

SQLite is used as a means to store information generated by the application.

### Communications Interfaces

The application will not be communicating with Internet explicitly. The only case of implicit connection is through the GPS system which is highly dependent on the Android device, Operating System and User settings. As the connection is employed by GPS, the application remains agnostic of it.

Additionally, the application will be communicating with SGshell through Bluetooth® in order to gather the relevant information. Said communication will be established by the OS and managed through the provided API.

### Memory Constraints

The amount of RAM needed should be around 256Mb of RAM to fit most phones in terms of RAM and phone storage at most 512Mb for database and the app. Meanwhile the app is at most 20Mb. But during the development while gathering the data there may be some adjustments on the memory consumption.

### Site Adaptation Requirements

In order to keep the installation easy and simple and because it's an Android app the main Requirement is to have the Play Store installed. Also it is required to have the SGshell and Bluetooth® connection for the application to function properly.

## User Characteristics

Users come from various educational levels without a necessarily high technology knowledge, therefor the application must be simple and easy to use. At the same time every hang gliding pilot passes a specific exam to get a license in order to fly. That means the users will be familiar with the hang gliding terminology, weather and the knowledge for flying at different flying sites.

## Constraints

The power consumption is a constraint because some flights can last for hours and the phone battery must handle the flight.

Another constraint is the amount of data gathered and their real time management, because a lot of information will be gathered from the device sensors and will be processed in real time.

## Assumptions and Dependencies

There is a possibility in the future to add an option to connect to social media, so users will be able to share their flights with their friends or upload their flight record to a global flight database. Also, based on the location, the application will be able to synchronize with a global flight database and check which flights have a conflict with the user’s flight to warn them in advance or during the flight. (future version)

# Specific Requirements

## Software Product Features

### Feature 1: Calculate elevation/altitude

#### Purpose

Users should be able to check, from the record of their flight, the altitude, for any part of the flight route, and the maximum altitude they reached.

[Flight Mode] The user must be able to check the elevation/altitude at any time, in order to decide if they should descent or ascent.

#### Associated Functional Requirements

##### Functional Requirement 1

###### Introduction

The elevation value is one of the most important features, since users of the hang glider have to know how they should act in order to reach the desirable location. Ignorance of this value sometimes can be fatal for users, because there is a maximum height and if they overcome that height they will have to deal with shortage of oxygen.

###### Inputs

The input values will be send to the application from a barometer, which is included in smartphones. It must be calibrated before each flight. (see reference [18])

###### Processing

We don’t really need processing since the barometer is responsible for this job, however we must make sure that the user knows that it should be calibrated before each flight, in order to record correct data.

###### Outputs

The altitude will be displayed in feet or meters based on user’s preference. The user can check the altitude in the post flight statistics or from the flight record.

[Flight Mode] The application will check the value of the barometer and display warnings in case there is a danger involving the pilot. The altitude should be displayed to the phone’s screen and should be updated every second.

### Feature 2: Calculate rate of descent or climb

* + - 1. Purpose

Humans are not able to sense climb and sink rates (unlike birds or other flying animals). In the past (before the invention of the variometer) sailplane pilots even if they could readily detect abrupt changes in vertical speed, their senses did not allow them to distinguish sink from lift, or strong sink from weak sink. They really had a hard time in the past since the actual climb/sink rate could not even be guessed at, except if there was a fixed visual entity or reference nearby. A fixed reference, like a hillside or the ground, is generally very unprofitable position to be in, for a glider pilot. The most useful forms of lift, like thermal and wave lift, are usually found at higher altitudes and it is very hard for a pilot to detect or exploit them without the variometer. (see reference [4])

* + - 1. Associated Functional Requirements
         1. Functional Requirement 1

Introduction

In order to calculate rate of descent or climb a variometer will be used. In hang gliding the variometer is used almost continuously, sometimes it also has audible output, during a normal flight to inform the pilot of rising or sinking air. Usually gliders are equipped with more than one type of variometer. There is a simple type of variometer which does not need an external source of power (which can save up power). On the other hand the electronic type, which needs a power source to be operative during the flight, can provide audio signals (for guidance). This instrument is not used during launching or landing, with the exception the case when the pilot wants to avoid releasing in sink (aerotow).

Inputs

The input values will be send to the application from the accelerometer.

Processing

Based on the data from the accelerometer we can calculate the rate of ascent or descent.

Outputs

User will be able to check from the flight record the rates for descent or climb that were encountered during their flight.

[Flight Mode] The application will check the value of the accelerometer, do some calculations and display warnings in case there is a strong lift or sink wave. The descent or climb rate should be displayed to the phone’s screen and should be updated every second. Descent and climb rate must be displayed in meters per second or feet per second based on the user’s preference.

### Feature 3: Calculate the position with GPS

* + - 1. Purpose

GPS (global positioning system) and a compass can be used to help the navigation during a flight. For competitions, GPS is used to verify if the contestant has reached the required check-points or not.

* + - 1. Associated Functional Requirements
         1. Functional Requirement 1: GPS

Introduction

The GPS can be used to visualise the location of the pilot on the map and assign some check points (for competition purposes). It allows users to check their current position and help them pinpoint a landing zone nearby. GPS can also be used to track and record the flight of the glider pilot.

Inputs

The input signals will be send to the application from the mobile’s GPS.

Processing

The flights will be recorded and saved on the mobile’s memory until they are deleted (or uploaded to a database/social media, which requires internet connection and an extended version of the application).

Outputs

The application will save the flights to a database on mobile’s memory and users will be able to inspect, rename or delete them.

[Flight Mode] The pilots will be able to see a map with their current location and the direction they are heading. During competitions pilots will also see some waypoints on their map so that they will know which route they should follow.

* + - * 1. Functional Requirement 2: COMPASS

Introduction

The compass is also useful for pilots as it helps them to understand to which way they are heading.

Inputs

The input for the compass will be send to the application from the mobile’s magnetometer.

Processing

The phone will do the processing for the compass calculations.

Outputs

[Flight Mode] The compass can be presented on the application’s interface with an arrow pointing to North.

### Feature 4: Temperature

* + - 1. Purpose

The temperature, even if it might not seem very important, it can affect the flight. Consider that when a glider pilot lifts higher in the sky the temperature around him/her goes down. So, it is important to keep track of the temperature. Also temperature can affect downdrafts and updrafts.

* + - 1. Associated Functional Requirements
         1. Functional Requirement 1: Temperature

Introduction

A thermometer can be used to send temperature values to the application for statistic purposes.

[Flight Mode] It can warn users in case it spots extremely low or high temperatures.

Inputs

The input will be given to the application from the thermometer.

Processing

No processing required, except for converting Celsius degrees to Fahrenheit degrees.

Outputs

Users will be able to check the minimum, maximum and average temperature of a flight.

[Flight Mode] The temperature can be displayed on the screen periodically. In case of extreme temperatures the application will warn the pilot. Temperature will be displayed in Celsius or Fahrenheit degrees.

### Feature 5: Humidity and pressure

* + - 1. Purpose

Temperature, humidity and pressure can affect the density of air. The density of air is one of the main factors which determine if the wing will fly faster or slower. Pressure can also affect the pilot’s health if it gets too extreme. Humidity and pressure can indirectly affect the flight, so we must make sure that those values are being monitored by the application. Also, sometimes we might be able to predict a cumulonimbus (also called thundercloud) based on humidity and temperature, so that we can warn the pilot to take the appropriate measures. (see reference [14], [15], [16])

* + - 1. Associated Functional Requirements
         1. Functional Requirement 1: Humidity

Introduction

A humidistat can be used to send humidity values to the application for statistics of flight records.

Inputs

The input will be given to the application from the humidistat.

Processing

Humidity can be used to calculate density of air.

Outputs

Humidity values will help the pilot to get a better understanding of the weather and the “nature” of the air that surrounds him/her during the flight. This will improve his experience and knowledge for the next flight.

[Flight Mode] The humidity can be displayed on the screen periodically. In case of extreme humidity values the application will warn the pilot.

* + - * 1. Functional Requirement 2: Pressure

Introduction

A barometric can be used to send pressure values to the application for statistic purposes.

Inputs

The input will be given to the application from the mobile’s barometer.

Processing

Pressure can be used to calculate density of air.

Outputs

Pressure values will help the pilots to combine their feeling during a flight with the pressure they were experiencing, at a specific part of the flight, giving them more experience for the next one.[Flight Mode] The Pressure can be displayed on the screen periodically. In case of extreme pressure conditions the application will warn the pilot.

### Feature 6: Calculate velocity

* + - 1. Purpose

During a flight velocity is very important for the pilot. If the velocity is below the minimum value, under certain circumstances, it might cause loss of altitude and a sharp fall of the wing. Velocity is also important during take offs and landings.

* + - 1. Associated Functional Requirements
         1. Functional Requirement 1

Introduction

For a given rate of climb there is an optimal speed to fly your particular glider (according to MacCready). This is the most efficient speed, if we take into consideration the balance between getting there fast to exploit good climbing potential, but not so quickly that you lose (valuable) height, which takes too long to regain. MacCready has calculated how that speed must be varied to compensate for associated lift and sink en route, and made flights a lot easier for pilots. (check reference [13])

Inputs

The input will be given to the application from the GPS and the accelerometer.

Processing

The velocity will be calculated based on GPS position and acceleration value.

Outputs

Users after the flight can check their velocity for a specific part of the flight and see their highest or lowest speed and when they reached those speeds.

[Flight Mode] The velocity must be displayed on the screen all the time. It should be updated every second. It can warn the pilot if the velocity drops below the minimum value. Velocity will be displayed on meters per second or kilometres per hour, based on user’s preference.

### Feature 7: Timer

* + - 1. Purpose

During a flight competition most of the time we need the time duration of a flight to decide who is the winner.

* + - 1. Associated Functional Requirements
         1. Functional Requirement 1

Introduction

All phones have a timer (clock) which can be used to calculate the time between take off and landing, or between two checkpoints. This information will also be added to the record of each flight. And maybe we can create a leaderboard with the best times (based on the same geographical checkpoints). Other pilots will be able to download the record from the flight, get the checkpoints and maybe try to beat the time record.

Inputs

The input will be given to the application from the mobile’s clock.

Processing

Calculate the time between two checkpoints and save it to the flight record. Also calculate the total time of the entire flight duration.

Outputs

At the end of the flight you will be able to check the flight time, and the time between each pair of checkpoints.

### Feature 8: Upload/Download flight’s information (Future Plan)

* + - 1. Purpose

During a flight the pilot must be sure that he/she is not entering an other’s flight route, to prevent accidents. How can the pilot know all the flights in a specific area? The global flight databases have these information stored on their servers, waiting for somebody to check them out.

* + - 1. Associated Functional Requirements
         1. Functional Requirement 1

Introduction

The pilot will be able to download the flights that are happening in a specific area, at a specific time and the application will warn him/her if he/she is heading to a route of another flight. Also there are areas that flights are forbidden for various reasons which will be marked also on the application, so that it can warn the user in advance.

Inputs

The user will be able to upload his/her flight record to a global flight database (like Leonardo) so that he/she can free some space from the mobile phone’s local memory.  
[Flight Mode] The input will be given to the application from a global flight database.

Processing

Upload a flight record to a global flight database or download a record to check the statistics.

[Flight Mode] Calculate the distance between user’s location and forbidden area and warn him (100m before).

Outputs

A flight record (if the user decides to download one from the global flight database).

[Flight Mode] The application will pop up warnings in case the pilot is heading to a route where at the same time another flight is happening.

### Feature 9: Microphone input/output

* + - 1. Purpose

During a flight the pilots are not able to fly with only one hand for more than a few seconds. A microphone combined with voice commands would be very helpful and more functional for them.

* + - 1. Associated Functional Requirements
         1. Functional Requirement 1

Introduction

All mobile phones have a Bluetooth® connection feature which can be used to connect the phone to a headset, which must include a microphone so that the pilot can interact with voice commands. Also the warning messages can be sent to the pilot with voice messages or warning sounds.

Inputs

The input will be given to the application from a microphone (headset) connected to the mobile device (with a 3.5mm headphone jack wired or Bluetooth® wireless connection).

Processing

Google libraries will be used to transform voice commands (speech) to text form and text form to voice messages or warnings. (check reference [19])

Outputs

Based on the voice command given by the pilot the application should display or respond to it with a voice message.

### Feature 10: Flight Records – History

* + - 1. Purpose

Users need access to their flight statistics for every flight in the past, in order to evaluate and compare their performance with previous flights. A History with flight records will be a very useful tool to keep track of their performance.

* + - 1. Associated Functional Requirements
         1. Functional Requirement 1

Introduction

In order to provide to the user the statistics of all flights, the data for each flight must be saved in a database, along with the date, time, duration of the flight and an identification name that the user can set. Every record will appear at the “History” screen in chronological order.

Inputs

Users simply click on a flight to see more details.

Processing

The program loads the information from the database.

Outputs

A new screen appears with all the statistics for the certain flight (statistics from previous features).

### Feature 11: UV index

* + - 1. Purpose

Users will be able to check the UV index during the flight as an additional information about the environment of their flight. (see reference [20], [21])

* + - 1. Associated Functional Requirements
         1. Functional Requirement 1

Introduction

In order to calculate the UV index the user must have the sensor and be connected on it. Additionally in order to get accurate metrics, the placement of the sensor must avoid shady surfaces.

Inputs

The input will be provided by the sensor which will be connected the device.

Processing

There will be some processing of the data in order transform the light wavelength into the UV index.

Outputs

A window displaying the UV index and some information about the value calculated.

## Performance Requirements

At any time only one user can have access to the application (on one mobile device).

Recorded flights should be stored to a file on the mobile device (if there is enough space), since an internet connection requires a lot of power and we do not want to leave the pilot in the middle of the flight with no battery. We can add a power bank (for backup) and warn the pilot in advance when there is little power left.

The application will require at least 256MB of RAM and 512MB of secondary storage to be able to run efficiently.

Altitude, velocity and climb/descent rate must be updated every second (without delays).

Humidity, temperature and pressure can be displayed periodically every 3-5 seconds.

The GPS and compass should change dynamically (as a normal GPS would on ground movement) during the flight and update location every second.

## Software System Attributes

### Reliability

SGshell provided by SignalGenerix is considered reliable. However, like any other hardware interface, its reliability is a function of the manufacturing process and the operational time. Also, the GPS system is considered reliable.

Light Gliding is dependent on the mobile’s operating system. Thus, any failure on behalf of the operating system would result in abnormal termination of the application. None actions are taken to restore the application’s before-the-failure state. A database file in SQLite though, is the only component that can roll back, even if the application crash in the middle of a transaction. In this way, information loss is prevented. (https://www.sqlite.org/howtocorrupt.html)

The statistics presented to the user are based on the measurements gathered from the Light Gliding, so they reflect the real “big picture” of the flight.

### Availability

Light Gliding is available on any mobile device with minimum Android version Marshmallow 6.0 and has access on PlayStore.

Also, power consumption is a constraint that defines availability.

Regarding user interfaces, there are some constraints. For example, when the user chooses to record a flight, he/she cannot navigate through the other screens. Moreover, maybe there are cases that an interface screen may not respond while others do.

### Security

Light Gliding’s database stores information and statistics about the user’s flights. These are considered personal data that should be kept private. Only the user has the authorization to share this data.

Furthermore, storing data in the database should be followed by the sanitization of the input. In this way it is not only ensured that user’s data is properly stored but malicious code injection is also prevented. (check reference 17)

### Maintainability

Maintainability phase includes:

* Bug fixes
* Future updates

Software bugs such as errors or faults in the system that cause incorrect or unexpected behavior should be resolved successfully.

Adding extra features or improving the performance of an existing one, might be a future update.

### Portability

Light Gliding is usable on Android phones and tablets only.

## Logical Database Requirements

The system will be using SQLite. A database stores:

* Elevation/Altitude (Feet or Meters)
* Temperature (Celsius or Fahrenheit)
* Humidity (%)
* Rate of descent or climb (feet per minutes or meters per second)
* GPS positioning
* Atmospheric pressure (N/m2 or Pa)
* Density of air (kg/m3)
* Velocity (m/s)
* Flight duration (minutes or hours)

The user has access to all the past flight records in the database, so he/she can view (or share with a future version) but not edit them, except the name of the flight record. A flight record includes all the listed information above. Flight records are organized based on the date and time they were recorded.

The application is responsible for gathering the data from the sensors and store it in the database using the appropriate format for each category of measurements.

## Other Requirements

# Appendices