University of Groningen

Software Architecture

Group 3

Smart Flood Monitoring

Authors:

Eedema, Gerrit
Putra, Guntur
Fakambi, Aurélie
Schaefers, Joris
Klinkenberg, Guus
Brandsma, Jeroen
Menninga, Wouter

Version: 0.2.3
24 Sept, 2015

Authors

Name	E-Mail
Eedema, Gerrit	G.Eedema@student.rug.nl
Putra, Guntur	G.D.Putra@student.rug.nl
Fakambi, Aurélie	A.Fakambi@student.rug.nl
Schaefers, Joris	J.Schaefers@student.rug.nl
Klinkenberg, Guus	G.Klinkenberg@student.rug.nl
Brandsma, Jeroen	J.T.Brandsma@student.rug.nl
Menninga, Wouter	W.G.Menninga@student.rug.nl

Revision History

Version	Author	Date	Description
0.1	all	06-09-15	Setting up the working environment and the initial document structure. Draft versions of the first four chapters created, which are (respectively) Context, Architectural business information, Requirements and Analysis.
0.1.2	Menninga, ...	10-09-15	First version of functional requirements
0.1.3	Joris & Eedema	13-09-15	Changed the context and enhanced the business information
0.2	Brandsma	13-09-15	Updated the requirements part
	Guntur	13-09-15	Updated the stakeholder of requirements part
0.2.1	Menninga	17-09-15	Updated the functional requirements
	Brandsma	17-09-15	Updated architectural vision and use-cases
	Klinkenberg & Eedema	17-09-15	Updated technical requirements, some formatting, started work on the first two decisions
0.2.2	Menninga	19-09-15	Added and updated functional requirements
0.2.3	Menninga	20-09-15	Updated assumptions and improvements to structure of analysis chapter. Added decision about type of water level sensor.
0.2.4	Brandsma	20-09-15	Updated requirements chapter. Added decision about cloud computing.
0.2.25	Eedema	19-09-15	Processed feedback
0.2.25	Eedema	19-09-15	Improved business chapter
0.2.4	Schaefers	20-09-15	Improved business chapter and processed feedback. Checked the use cases and put them in tables.
	Putra	19-09-15	Processed feedback, revised the stakeholder, and added decision on database selection.
0.3	Menninga	23-09-15	Improved functional requirements after feedback
	Brandsma	24-09-15	Revised architectural vision, extended and improved use-cases
	Menninga	24-09-15	Changed layout of the risk assessment section
	Fakambi	24-09-15	Improvements to the non-functional requirements
	Eedema	24-09-15	Improvements to the business part
	Menninga	24-09-15	Changed layout of the risk assessment section
	Putra	24-09-15	Improved stakeholder with new quality attributes. Added key-drivers after stakeholder.

Contents

List of Figures

List of Tables

1 System Context

We are the architecture team of the company RugSAG3. Our team is specialized in making smart systems for environmental monitoring.

All around the world, natural disasters cause a lot of trouble. These disasters can result in catastrophic events that cause deaths and require a huge amount of money to repair the damages caused. People lack the knowledge on when the disaster is about to happen, how to properly prepare and how to properly act during such a disaster.

Climate change and extreme weather phenomena cause these disasters to get worse over time. Every year the amount of these disasters increases and they become increasingly more severe. This causes the damage of the natural disasters to increase, which means that there is a great need for a system that can reduce this damage and helps the people during natural disasters like these. The system that we develop aims to do this. The goal of our first product is to:

1. safe lives,
2. reduce damage costs,
3. reduce the social consequences.

To achieve the aforementioned goals, the system will provide predictions with regard to upcoming floods.

The first release of this system that we, as the architecting team of RugSAG3 design, will only support floods as a natural disaster. The system will provide warnings and guidance to the necessary people before and during a flood. By using various kinds of sensors, like vehicles and control units, this system monitors certain areas. If something suspicious happens, the system will check and verify the information in order to not give false flood warnings. When a flood indeed occurs (or will occur), the system provides warnings and guidance to the necessary people during and, if possible, before a flood.

When this flood monitoring system works as planned and shows that it can indeed reduce the damage of floods, RugSAG3 will extend the systems functionality by adding support for the monitoring of other kinds of natural disasters and situations that desire a monitoring system.

Over time, the system will reduce social, financial and human losses of natural disasters, starting with floods. This will revolutionize the way we think about natural disasters. Though the market around environmental monitoring is very competitive, this system aims at being a core part of future monitoring systems by being as dynamic and flexible as possible. Thereby allowing new features to be added easily, allowing the system to grow over time, including new upcoming user needs.

2 Architectural business information

The following section describes the different aspects of the business environment of the Smart Flood Monitor. First we will explain our vision and why there is place for us at the market.

After this the product and its stakeholders will be explained. This chapter is completed with a more detailed look at the business model and some models about the market and the financial prospect.

2.1 Business opportunity

There are many natural disasters happening each year all over the world. Each year these disasters take lives, destroys a lot of properties, and cause social disturbance. Looking at, for example, the Indian ocean's tsunami in 2004, it looks that the damage could have been significantly reduced if the necessary people were warned.

In the future more floods are expected because of global warming. The rise of the sea level is a consequence of global warming. Another consequence is the increase of extreme weather events, for instance heavy rainfall. It is expected that natural disasters will cause $300 billion in losses annually in the upcoming decade. This justifies to invest a high amount of money to minor the losses of a flood. Not only the losses in terms of money, but more importantly lives, and also social impact.

A next generation reliable flood monitoring and warning system will help lower the catastrophic impact of floods. The system will imminent floods and send out people in order to reduce the impact of a flood. In the future the system should be able to monitor other kinds of disasters and send out warnings.

The Netherlands is a country that is situated for large parts under the sea level. Using dikes and other solutions they protect their country against the water. The increasing sea level causes extreme danger in the Netherlands. It is important that they can monitor how dangerous imminent floods are and take action if things are looking to wrong. If things go wrong people within the area should be warned in order to take action and safe their self, others, and their belongings.

2.2 Mission statement

A flood monitoring and warning system helps us to enforce our vision, to limit the social and financial consequences of floods and reduce the loss of human lives.
The Smart Flood Monitor will cause a revolutionary innovation on the environmental monitoring market. RugSAG3 will be able to offer a system that can detect floods early and correctly by using the newest available techniques. At first the system will be build for monitoring floods, but in the design there is taken in to account to monitor other kind of natural disasters.

2.3 Business Rational

Our unique selling point is to provide a system with a low selling price and high profit from the service contracts and upgrades. However, RugSAG3 is not the only player on the environmental monitoring market as there are a lot of other competitors taking part on this area of expertise. This means that it is important to evaluate the strengths and weaknesses of RugSAG3. Using this info, we can reflect ourselves with respect to the competitors, leading to a better understanding of the opportunities and threats RugSAG3 has in the market. This will increase our position in the market and further enables RugSAG3 to play a leading role in future environmental monitoring systems.

We map our position on the market by using SWOT-analysis. This analysis maps our strengths, weaknesses, opportunities, and threats. The results of our analysis is shown in the table below.

Strengths	Weaknesses
• Adjustable system that is future proof • Having a low selling price. Lower then the competitors • Diverse team with several skills in IT and management • Having a good management team • Frequent discussion with technical and business experts in the field. • Experience with working with sensors	• Decisions need to be made by the entire project team. There is a higher chance of a difference of opinion is members, holding back the project and leads to time wastes • No experience with creating environmental monitoring systems • No knowledge of floods • Some main features rely on country-specific systems
Opportunities	**Threats**
• Due to climate change, the market will grow and such a system becomes more urgent • Diverseness of cultures in the development team • Allowing the system to be flexible so it can also be used for other kinds of monitoring activities • Smart sensors are a hot topic, new sensors will be developed. Making the system support and use the newest sensors allows it to obtain more, and a wider variety of valuable information	• High production costs • Competitors will enter the market because of growing market • New competitors will enter the market • Climate change will forces the system to be improved over time. • Changes of the external systems that our system uses to properly communicate. • The sizes of certain area's to monitor can be too big to get a good and reliable view of them

2.4 Business rationale

RugSAG3 will develop a new flood warning system in order to minimize the damage caused by a flood. As said before, in the Netherlands protection against floods is an important issue, because a major part of the Netherlands itself is actually below sea level. Global warming will increase the urgency of this issue. The people within the Netherlands must be well protected against floods. The flood warning system will detect floods, warn people and governmental institutions located in the disaster area and provide guidance where to go to. This visualized in the figure below.

RugSAG3 is new within the flood warning system market. By using sensors and automated systems, a reliable and adequate product that uses the newest technologies will be launched. RugSAG3 will use hardware which is already on the market and is tested. Buying third party hardware will also speedup the development of the system and lower the costs.

The product price will be low in order to get a market share and prove the product in a real-time environment. By providing maintenance and updates in the future RugSAG3 will earn money to improve the product further and sell it to other potential customers. This in combination with the increasing need of a reliable warning system for imminent floods will result in a viable business.

The unique selling points of our system are:

- To provide a system with low selling price and high profit from the service contracts and upgrades
- Warn the people in and around the area as soon as possible if the system is very certain that a flood is or will happen
- Inform the necessary parties about the details of the flood so the right preparations are made.
- Inform people how to save themselves, how to save other people or how to save valuable goods

This way, people will know when an area gets flooded, or when it is about to happen. These are unique features for our system, which will eventually make the system successful. The main goal of this project, however, is to safe lives, reduce costs and reduce social consequences. These main goals will be met when:

1. 80% of the people in a dangerous area regarding a flood, receive a warning message. This message must

contain enough information for receivers to know whether they are save or not and if not, how they can get to a save location.
2. 80% of the people who receive a warning successfully get to a safe environment in time.
3. 80% of the people receiving information before or during a flood, find these messages helpful and reported that it guided them successfully in order to save extra lives and/or goods.

When the first version of Smart Monitoring is released and is used to start monitoring actual floods, its success will be measured according these statistics. Getting a warning message to the people who requested to be warned is the most important thing to do. Using this warning message, people can move to a saver location.

2.5 Product and service description

RugSAG3 offers a flood warning system. When a imminent flood is monitored by the sensors of the system a warning should be sent to governmental organizations and people within the danger area. Also a possibility of guidance should be provided when a flood is happening to assist rescuers and guide inhabitants to safe areas.

Basically the system will consist of four subsystems: monitor the state of dykes and water levels, analyze the data from the monitoring part to detect imminent floods, warn governmental organizations and inhabitants in the danger area, and provide guidance to search and rescue organizations and inhabitants.

The first subsystem (monitor the state of dykes and water levels) will consist of various sensors that are placed near and in dykes and water ways. The state of the dikes must be monitored continuously, i.e. pressure of the dyke. Also, sensors must be installed to monitored continuously the water level. The data of all the sensors will be sent to a server, in a safe location, to store all the data.

The second subsystem (analyze the data from the monitoring part to detect imminent floods) will analyze all the data from the sensors and data from weather forecasting service. Based on this data an algorithm will monitor continuously if there are dangerous situations.

The third subsystem (warn governmental organizations and inhabitants in the danger area) will send warning messages when the algorithm identified a dangerous situations. The safety region will receive a warning message that an area is in danger. Information like position, area, sort of danger and amount of danger will be send. The safety region will be responsible to take action based on this information. Inhabitants can receive warnings via sirens, mobile phone, radio, television, and by UAV.

The last subsystem (provides guidance to search and rescue organizations and inhabitants) It will help people that are in a dangerous area by sending them information on how to rescue themselves as in a save way. Rescue organizations receive information about the area's that are likely to have casualties in them and it informs them on how to get to those locations save.

The service will consist of maintenance for the product and upgrades.

Floods will be the first natural disaster the initial system supports. The initial system sends warnings to the people in danger and to emergency services, but will not yet interact much with the user. However, the system doesn't stop there and will get increasingly more capabilities. The extra capabilities of system include:

- Support for more kinds natural of disasters
- More individual guidance
- Interaction with the system, users can give input
- Support more sensor
- Support of multiple communication networks to send information

The first thing RugSAG3 will focus on after the first release, is increasing the ability for the users to interaction with the system. This allows the system to provide more personal help to the users and increases knowledge the system has of the area. However, mistakes can easily be made by a user and so this data needs to be verified in order to get valuable information. That is why adding this feature goes beyond the scope of this project.

Adding sensor support is a continuous process. The sensors that are available are steadily increasing and their technology becomes more advanced. RugSAG3 will monitor the sensor technologies and improvements to check if these can improve the system. Depending on where the system resides, it will need to interact with various kinds of networks en media. Initially the flood warning system will focus on communicating with resources in the Netherlands. However, if a future release implements additional support for monitoring tornadoes or volcanoes, the system will most likely not be in the Netherlands.

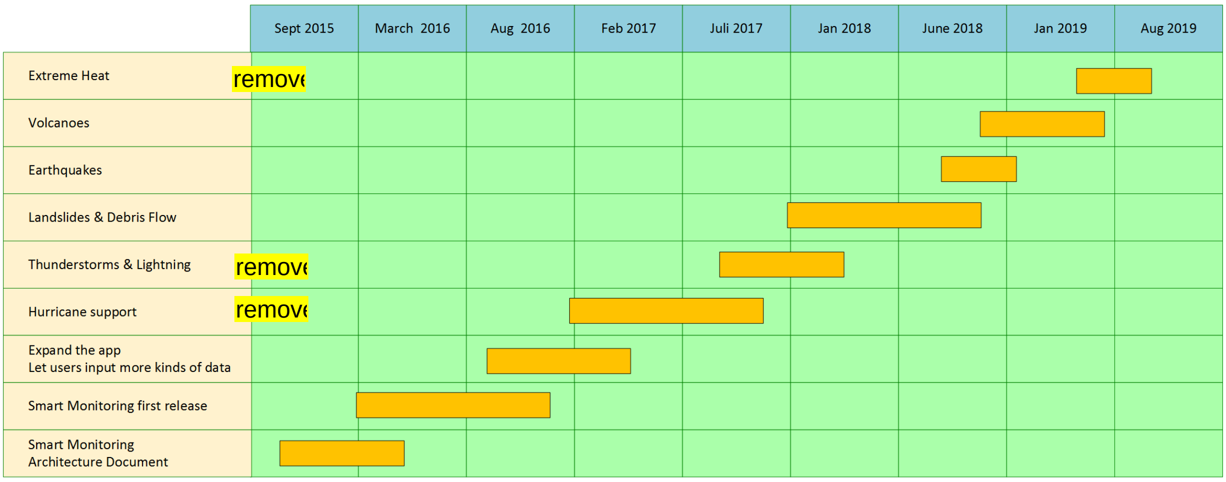

Figure 2.1: Future releases.

2.6 Target audience

The target audience is where we focus on to market our product. The Dutch Ministry of Infrastructure and the Environment is responsible for floods in the Netherlands. They make the decisions to buy the system.

The Netherlands is divided into several safety regions. A safety region consists of all the emergency services, municipalities, water boards. They make decisions about how to make their region safer. All those organizations should be seen as the target audience. They must see the benefit of using RugSAG3's Monitoring and Warning system.

2.7 Financial model

The financial model will be a low product price. This in order to price the product low in the market. A service description for maintenance will be offered. Also updates will be sold to the customer

2.7.1 Software Architecture costs

The software architecture team of RugSAG3 consists of six members. Creating the architecture of the project is estimated to take ten weeks. All team members will spend 15 hours a week on the project. This totals $6 * 10 * 15 = 1050$ working hours. Each working hour costs €150,-. Total spend on the software architecture is €157.500

2.7.2 Development costs

The development costs are calculated in the table below.

Description	**Man hours**
Get values from various sensors	160
Get weather forecasts	160
Flood prediction algorithm	3100
Warning messaging system	2000
Guidance information system	1500

Redundancy and fail over systems	1000
Testing & debugging	600
Release build	250
Overhead	1000
Total hours	9770

The above table shows the project needs approximately 9770 hours to develop the system. The development is done by RugSAG3 self. Each member of the project team assigned to develop this system is paid €50 an hour. This results in a development cost of $50 * 9770 = 488.550$. As shown in the table below.

Hours	9770
Cost per hour	€50
Total cost	€488.500

2.7.3 Hardware costs

Around 17,000 kilometres of dikes protect the Netherlands against flooding [1]. Sensor costs include:

- technical costs of the sensor
- implementation costs
- system implementation costs

The system needs:

- Sensors
- State monitor server
- Danger check server
- Warning system server
- Guidance server

[2] pp 127. The first case considers monitoring of the smallest stretch of 6 km with a simple monitoring system S1. The monitoring system consists of MEMS sensor modules (e.g. GeoBeads) with a claimed life time of 10 years, which are installed with conventional CPT push-in techniques. Three sensors are installed per cross-section and a cross-section is installed every 100 m. The total installation costs of monitoring system S1 are,

The system uses dyke meters and water meters. The system needs at least 200 water meters and 75 dyke meter. A water meter costs about €5 and a dkyke meter costs €60 each. Each server costs €1.500. This results in the minimum hardware costs of be €11,500.
However, it is very important that the system doesn't fail. So if a few components or sensors fail, the system should still be fully functional. To make sure the system won't fail if a component fails, the hardware is redundantly set up four times.
The total hardware costs then results in: €46.000

2.7.4 Total costs

The total costs for the system is calculated in the table below.

Software Architecture costs	€157.500
Development costs	€488,500
Hardware costs	€46.000
Total cost	€692.000

So the total costs for developing the system is €692.000

2.8 Competitors

Siemens is a competitor that already has developed a flood-warning system in Belgium in 2006. When the system detects a imminent flood it sends a SMS to people that live near the rivers in order to warn them. The system is implemented in a small region, which consists of only 3 rivers. The total cost for this system were €230.000. Later on they continued engineering the system. The system uses sensor which measures: temperature, water pressure, and shifting. They participated with this system in the Urban Flood project from the EU. The strong point of this competitor is that they already have funding and also experience with building such a system. Siemens is the main competitor for RugSAG3. Their strength is that they already have a proven system in Belgium and have a lot of experience through participating in flood monitoring research projects. The opportunity for RugSAG3 is Engineering a flexible system that can be used in the future for other kind of disasters. Further on by selling the product at a low selling price and profit on the maintenance work and upgrades the Netherlands will be interesting in using the product.

Further there are Universities that do research on flood warning systems. At the Malaysian Institute Information Technology they developed a product prototype of a system called Intelligent Flood Information System via SMS. Waterlevel sensors are the only one they used. When their is a flood they send a warning SMS to people that are within the area. In the future this could become a competitor if they create a start-up or sell the idea to a company.

3 Requirements

This chapter will describe the vision and use it to derive stakeholders to be able to properly write use cases and stories. These will be used to extract functional, commercial, technical and evolution requirements. Afterwards, a risk assessment will take place, to ensure that the project is not at great risk.

3.1 Architectural vision

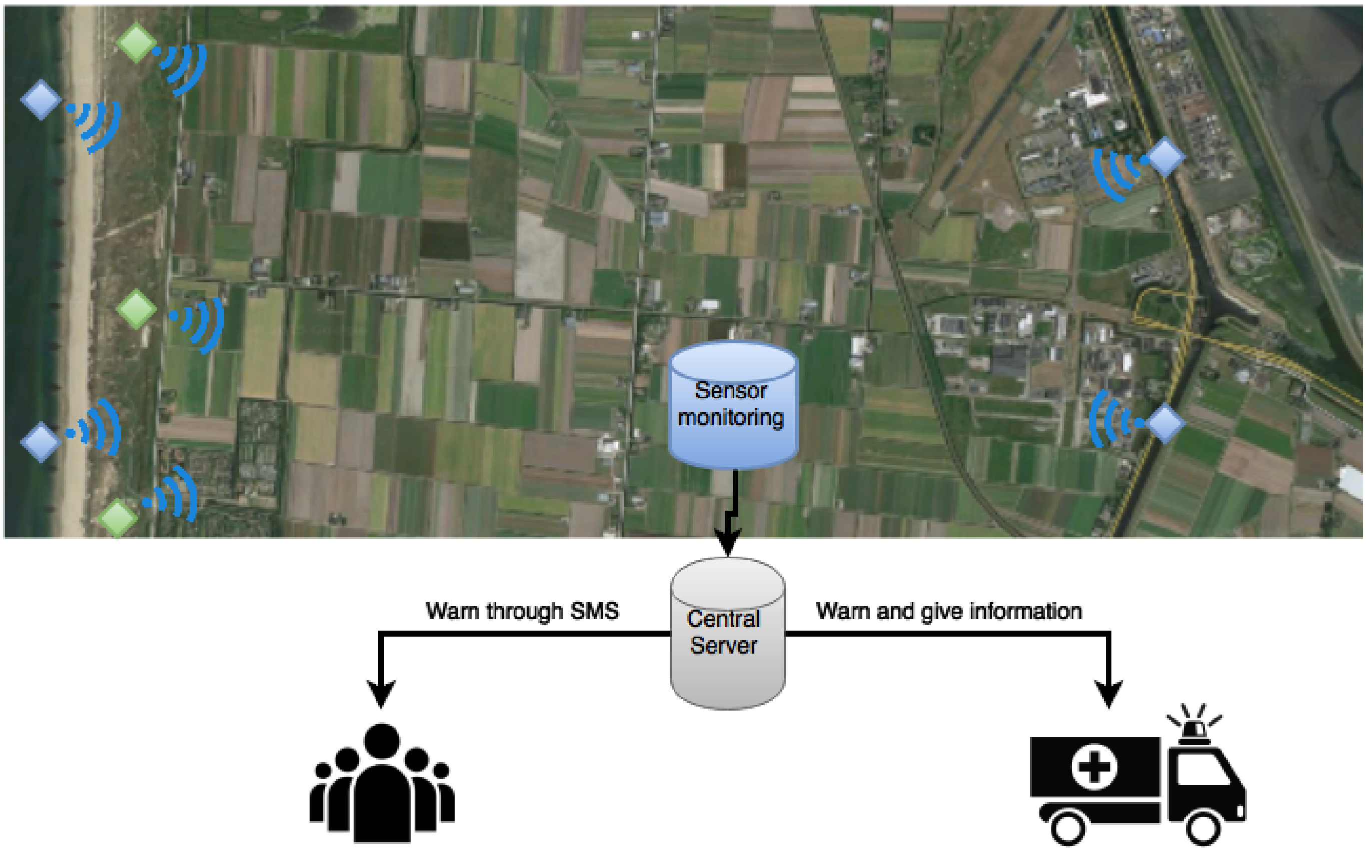

Figure 3.1: Schematic overview of the flood monitoring system

The smart flood monitoring system consists of multiple parts. These are represented in figure 3.1. First of all there is the monitoring part. This part monitors the current state of the environment. To achieve this, we need a lot of data. This data is obtained by sensors and weather data. We use sensors to get the current water level of waterways, these water sensors are shown in the figure as blue squares. We also measure the density and structure of dikes, this is done by pressure meters, temperature meters and tilt meters. These are represented as green squares. The data that is obtained through all sensors will be send to the central server. Here the information will be processed. The system then determines if there is an imminent flood.

In case of an imminent flood a warning will be issued to the government and the citizens who live in the threatened area. We do this by issuing a warning to the government. In their turn the government uses their infrastructure to warn the citizens. In the Netherlands this infrastructure consists of a siren system and an SMS-system. Besides this people can also apply for our SMS service. People who are subscribed to this service will receive a text message with more information about the (imminent) flood.

In case of a real flood the citizens who subscribed to our service will receive a MMS message with a route to a safe area. These MMS messages contain generic routes to get to a safe area. This means that when a flood happens, the same map is send to different users.

3.2 Stakeholders and their concerns

This section defines all stakeholders of our system and describe the concerns of the stakeholders. A stakeholder might be a person, group of persons, or organization that are involved in our system. There are eight stakeholders, ranged from first parties to third parties stakeholders. We use several quality standards from "Software Requirements" book by Microsoft [3]. Those quality standards are described in Table 3.1.

Table 3.1: Quality attributes of Software Architecture from "Software Requirements" Book [3].

Quality Attributes	**Brief description**
Availability	The extent to which the system's services are available when and where they are needed
Interoperability	How easily the system can interconnect and exchange data with other systems or components
Performance	How quickly and predictable the system responds to user inputs or other events
Reliability	How long the system runs before experiencing a failure
Security	How well the system protects against unauthorized access to the application and its data
Usability	How easy it is for people to learn, remember, and use the system

There are six quality attributes, as can be seen in Table 3.1, for measuring stakeholders' concern regarding our system. Furthermore, we also add profitability as another quality standard to improve measuring stakeholders' concern. Detailed description of stakeholders and their concerns are explained below.

Product owner is concerned about performance, reliability, and profitability.Product owner funds the whole project. Product owner highly concerns about the profitability. Thus, to gain big market share and extract large profit from this product, product owner has to make this product reliable. Furthermore, to compete with other competitors in this area, product owner also concerns about the performance of the system.

Developers are concerned about availability, interoperability, performance, reliability, and security. We, the architect team of RugSAG3 company, are also part of this. This stakeholder is responsible for the development of the systems until its ready for production. Including architecting, designing, analyzing, testing and implementing this Smart Flood Monitoring System.

Competitors are concerned about performance, reliability, security, and profitability. Competitors give negative effect on the system because competitors will be aiming on the same customer target. On the other hand, competitors are also triggering us to make a really good system in order to be able to compete with them and to save more lives. Thus, competitors must also be kept in consideration.

Government is concerned about availability, interoperability, performance, reliability, and security. Government will be the main customer of this product, specifically, The Dutch Ministry of Infrastructure and the Environment. Government will be part of mitigation when the flood is imminent. This system will help the government by notifying them when this system detects flood and what is the recommendations to do next along with some data regarding this system's findings.

Citizens are concerned about availability, reliability, usability. The Dutch residents are indirect user of this systems. However, they are also able to directly subscribe to this service and thus they want this system to be adaptable to their current technological viewpoint. Furthermore, they want this system to always be available and run correctly and notify them with the reliable information.

Insurance companies are concerned about performance, reliability, and security. The damages caused by flood sometimes are also covered by the insurance companies. Thus, the insurance companies will also be part of the stakeholders and they will make sure that their business is running well.

Local companies are concerned about interoperability, reliability, and usability. Local companies will also be affected by the flood, they also have a lot of resources which are in danger. Local companies want to know whether or not this system is reliable so that they can arrange a proper action sets when the flood comes to save their assets. Moreover, local companies are also willing this system to be as adaptable as possible to their current technological viewpoint.

Emergency services are concerned about interoperability, performance, reliability, and security. Emergency services are important when any accident happens, including flood. They will be really concerned about the thing that makes this system reliable, inter-operable to their current system.

Table 3.2 illustrates the stakeholder concern matrix. In our approach every stakeholders are equally the same. Thus, each stakeholder receives 100 points in total that has to be distributed among all the concerns.

Table 3.2: Matrix of stakeholders concern.

		Concerns						
		Availability	Interoperability	Performance	Reliability	Security	Usability	Profitability
Stakeholder	Product owner			30	30			40
	Developers	10	10	20	30	20		
	Competitors			20	30	20		30
	Government	20	10	25	25	20		
	Citizens	30			40		30	
	Insurance companies			30	35	35		
	Local companies		30		35		35	
	Emergency services		15	30	30	25		
	Total	60	65	155	255	90	65	70

As can be seen from Table 3.2, the most important concern of our system is the reliability, following performance as the second most important concern. This is also identical with our significant key driver.

3.3 Key-drivers

Table 3.2 has shown the concerns of the stakeholders of our system. Every stakeholder is equally the same in our system. Thus, each stakeholder receives 100 points to be distributed to all quality attributes. By doing this, important quality attributes can be determined and those will be the key-drivers of our system.

It is clear that **reliability** is by far the most important quality attribute of our system, followed by **performance** as the second most important, and **security** as the third most important quality attribute. Thus, those three quality attributes will be the *key-drivers* of our system. We decided to choose only three quality attributes as the key drivers for our system because choosing more than three may possibly drive our system architecture to be more complex and consume more time to be developed.

3.4 Stories and use-cases

This section will give an overview of the different use-cases. Figure 3.2 displays the use-case diagram. This provides an overview of the use-cases with their actors. In the subsections below, the architectural important use-cases are explained in more detail.

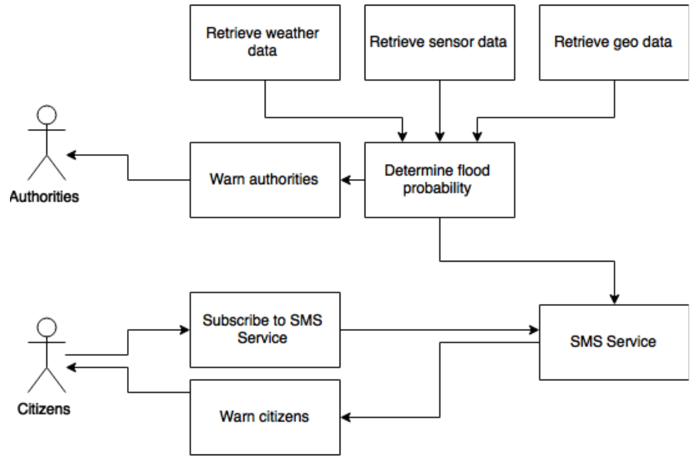

Figure 3.2: Use-case diagram

3.4.1 Retrieve sensor data

Number	UC-1
Description	The system receives data from the different sensors deployed
Stakeholders and interests	• **Developers**: Developers need to work with the sensor data
Primary actor	System
Scope	Monitoring part of the system
Level	Sub process
Precondition	The sensor is connected to a processing unit
Main success scenario	1. The sensor does a measurement 2. The sensor sends the data to a processing unit 3. The processing unit normalizes the received data 4. The processing unit sends the normalized data to the main processing node 5. The main processing node stores the data in the database
Postcondition	The system received and stored the sensor data
Alternatives	2a. The data can't be sent. Data will be lost The use-case ends
Related requirements	FR-1, FR-2, FR-3, FR-4, FR-5, FR-6

3.4.2 Retrieve weather data

Number	UC-2
Description	The system receives data from the weather forecast service
Stakeholders and interests	• **Developers**: Developers would like to have a simple to use API
Primary actor	System
Scope	Monitoring part of the system
Level	Sub process
Precondition	The system needs external weather data to predict floods
Main success scenario	1. The processing unit determines it needs forecast weather data 2. A call is made to the weather forecast service 3. The weather forecast service returns the requested data
Postcondition	The system received the forecast data
Alternatives	3a. The data can't be returned. Repeat this process with another weather forecast service. If none are available, proceed monitoring without weather forecast data. After 5 minutes try to reconnect.
Related requirements	FR-7

3.4.3 Citizens subscribe to the SMS service

Number	UC-3
Description	Citizens can subscribe to the SMS service, so when a flood happens they will get a direct text message
Stakeholders and interests	• **Citizens**: Citizens want to be warned as soon as possible.
Primary actor	Citizen
Scope	Warning part of the system
Level	Sub process
Precondition	Citizen has a mobile phone and is not subscribed to the SMS service
Main success scenario	1. Citizen sends a text message to our service 2. The SMS service receives the text message 3. The SMS service stores the phone number in the database 4. A text message is sent back to the citizen with confirmation
Postcondition	Citizen is subscribed to the SMS service
Alternatives	2a. The text message is not received The use-case ends
Related requirements	FR-17

3.4.4 Determining flood probability

Number	UC-4
Description	The central processing unit calculated the probability of a flood
Stakeholders and interests	• **Developers**: Developers have to work on this part • **Emergency Services**: Emergency services want to know when a flood warning is triggered • **Government**: The government would also like to know when a flood warning is triggered
Primary actor	System
Scope	Monitoring and warning part of the system
Level	Main process
Precondition	The sensor data is available
Main success scenario	1. The central processing unit gets the latest sensor data from the database 2. The central processing unit gets the latest weather forecast data 3. The central processing unit calculates the probabilty of a flood 4. The central processing unit stores the probability value in the database 5. The central processing unit determines that a flood is imminent based on the probability value 6. A warning is send to the emergency services 7. A warning is send to the government 8. A warning is send to the citizens
Postcondition	The flood probability is calculated and stored. If the probability exceeds a certain threshold, a warning is sent to the authorities and citizens
Alternatives	5a. The probability is not above the threshold The use-case ends
Related requirements	FR-8

3.4.5 Warn citizens in case of an imminent flood

Number	UC-5
Description	Citizens who are subscribed to the SMS service will be warned through text messages in case of an imminent flood
Stakeholders and interests	• **Citizens**: When they are subscribed, they want to warned in case of an imminent flood
Primary actor	Citizen
Scope	Warning part of the system
Level	Sub process
Precondition	There is an imminent flood and citizen is subscribed to the SMS service
Main success scenario	1. The processing unit sends a warning about an imminent flood to the SMS service 2. The SMS service composes a list with phone numbers to warn 3. The SMS service sends a warning to all phone numbers on the list
Postcondition	The citizens who are subscribed received a warning
Alternatives	3a. A message can't be sent to the citizen Wait a minute and resend The use-case ends
Related requirements	FR-18, FR-19

3.4.6 Warn authorities in case of an imminent flood

Number	UC-6
Description	Government and emergency services receive a warning about an imminent flood
Stakeholders and interests	• **Government**: The government wants to warn the citizens in case of a flood • **Emergency services**: The emergency services want to help the citizens in case of a flood
Primary actor	Government, Emergency services
Scope	Warning part of the system
Level	Sub process
Precondition	There is an imminent flood
Main success scenario	1. The processing unit determines what area will be under water in case of a flood 2. The processing unit determines how many people will be affected by the imminent flood 3. The processing unit predicts how the flood will develop in the following period 4. The processing unit will create a map based on the current state and predictions 5. The processing unit sends the map to the government and emergency services
Postcondition	A map with current and predicted data is sent to the government and emergency authorities
Related requirements	FR-10, FR-11, FR-12, FR-13, FR-14

3.5 Functional requirements

This section lists the functional requirements of the system.

Nr.	Prio	Description
FR-1	**Must**	The system is able to receive input from water level sensors. This information will be used to determine if there is an imminent flood.
FR-2	**Must**	The system is able to process input from water level sensors.
FR-3	**Must**	The system is able to receive input from pressure sensors of the dykes.
FR-4	**Must**	The system is able to perform an analysis for the pressure on the dykes based on the input from the pressure sensors.
FR-5	**Must**	The system is able to receive input from water level sensors of waterways.
FR-6	**Must**	The system is able to perform an analysis for the water level in the waterways based on the input from the water level sensors.
FR-7	**Must**	The system retrieves weather forecasting data from weather forecasting services, which consists of predictions about the precipitation and wind data. This is used by the system to help in determining when a flood becomes imminent.
FR-8	**Must**	The system is able to detect when a flood is imminent by combining the retrieved sensor data and weather forecasting data.
FR-9	**Must**	The system can retrieve geographic information (road data and terrain height data).
FR-10	**Must**	The system computes the area affected by a flood, in zones of 5 by 5 km, by using the location data of the sensors and geographic information.
FR-11	**Must**	The system is able to perform an analysis, resulting in an estimated expected water level for areas which are affected by a flood, based on the water level sensor data, geographic data and weather forecast information.
FR-12	Should	The system estimates how the water level in the areas affected by the flood will develop for every hour, up to 12 hours in the future.
FR-13	Should	The system can compute the number of civilians living in the areas affected by the flood.
FR-14	**Must**	When a flood is imminent, the system sends a warning to the emergency center. The warning contains information about the flood: the area affected by the flood, the expected water level in those areas, how the water level will develop in the coming hours and the number of civilians living in the affected area.
FR-15	**Must**	The system can compute a safe area, not affected by the flood, where citizens can be evacuated to in case of an (imminent) flood.
FR-16	**Must**	The system can construct a map, showing citizens safe routes to evacuate the area affected by the flood and how to get to a safe area.
FR-17	**Must**	Citizens are able to subscribe to flood warnings about imminent floods.
FR-18	**Must**	Citizens who are subscribed for flood warnings are warned about imminent floods by text message.
FR-19	**Must**	Citizens who are subscribed for flood warnings get a map with generic route information to a safe area by MMS. This is the map generated in FR-16.
FR-20	**Must**	The system can detect a faulty sensor, either when the sensor raises an error or when the data from the sensor is inconsistent with other sensor data.
FR-21	**Must**	There is a control panel, where maintainers of the system have access to.
FR-22	**Must**	The system reports faulty sensors, so they can be viewed in the control panel.
FR-23	**Must**	Warnings of the system can be viewed in the control panel.
FR-24	**Must**	Errors of the system can be viewed in the control panel.
FR-25	**Must**	The readings of the sensors can be viewed in the control panel.
FR-26	**Must**	The system can make backups of it's data (configuration data etc.).

FR-27	**Must**	The system can store created backups on a remote location.
FR-28	**Must**	The system can retrieve the backups it previously created.
FR-29	**Must**	The system can restore the backups it previously created.
FR-30	Could	The system is able to detect extreme weather phenomena, like storms etc.

3.6 Commercial non functional requirements

In this section commercial non functional requirements are presented.

CNFR-1 The system is affordable. The selling price of the system to the authorities is about *********** euros per city or municipality. This price is lower than 70% of the competitors price.

CNFR-2 The sensors have a good quality , the sensors companies have good ratings so we don't have replace the sensors often => less money spent on repairs. The guarantee of the sensors should be about three years.

IDEAS : **CNFR-** A video explains how the system works to the end-users : authorities and emergency services. **CNFR-** Advertissement ???

3.7 Technical non-functional requirements

This section describes the technical aspects that are important to the system as requirements. These requirements determine various APIs and programs that the system will rely on.

3.7.1 Reliability

Nr.	Prio	Description
REL-2	**Must**	Data from the sensors is sent via a TCP connection

3.7.2 Resilience

The system needs to be resilient to recover from errors and mistakes without impacting the systems functionality.

Nr.	Prio	Description
RES-1	**Must**	The system recognizes failures within half an hour
RES-2	**Must**	The system recovers from failures without the Quality of Service or the functionality of the system being affected.
RES-3	**Must**	The system continues to function with the same Quality of Service in a situation where up to 10% of the sensors suffer from failures.

3.7.3 Performance

Nr.	Prio	Description
PERF-1	**Must**	Data is transmitted from and to the system with a minimum speed of 10 megabits per second
PERF-2	**Must**	The data transmission between the sensors and the system is about 10 mB/sec
PERF-3	**Must**	The time for the system to calculate if there is a flood or no according to a critical level and the data received from the sensors is about one minute(?).
PERF-4	**Must**	The system should be 99.9 available.

3.7.4 Interoperability

Nr.	Prio	Description
INTR-1	**Must**	The system pulls weather forecasts from five weather forecasting services.
INTR-2	**Must**	The system pulls water forecasts from one water forecasting service.
INTR-3	**Must**	When the system detects a high risk of flood, it warns the government automatically using the government provided API.
INTR-4	**Must**	When the system detects a flood, it automatically tells emergency services where a flood is happening using the API provided by the emergency services.
INTR-5	**Must**	The system sends out a SMS to all users who are subscribed to flood warnings.
INTR-6	**Must**	The systems is able to easily connect to different sensors.

3.7.5 Security

Nr.	Prio	Description
SEC-1	**Must**	Access to the system is restricted to user accounts that are stored in a database.
SEC-2	**Must**	All communication to, from and within the system are encrypted.
SEC-3	**Must**	User account information are hashed using bcrypt after being salted with 128 randomly generated characters.
SEC-4	**Must**	The system is protected by a firewall that at least scans at the application layer, while also scanning for and preventing DDoS attacks.
SEC-5	**Must**	The system communicates with it's sensors via a REST API that only allows for HTTPS connection.
SEC-6	**Must**	All system data must be backed up every 24 hours.
SEC-7	**Must**	Backup copies are stored in a secure location which is not in the same area as the system (50 km).

3.7.6 Scalability

Nr.	Prio	Description
SCALE-1	**Must**	The database and services run in parallel on a private cloud that is hosted within the Netherlands
SCALE-2	**Must**	Cassandra is used as database for the storage of the sensor data
SCALE-3	**Must**	The system is configurable to run in different areas and with different sensors.

3.8 Evolution requirements

When establishing the project, architects of the system listed a certain number of requirements which describe the features of the system. But due to environement changes and end-users requests for example the requirements evolve.

ER-1 Changes in the display of the map
Evolution of FR-19 Citizens who subscribed for flood warnings get a map with specific route information (according to their location) to a safe area through the app/API (in association with Google Map).

ER-2 : Adding of external input Citizens can contribute to the guidance by giving extra information(for example if they identified a safe route near their location) but this information will be check by an operator (thanks to photograph by the UAV's for example).

ER-3 : New OS The system is able to work with others OS than Linux : Windows, MacOS.

ER-4 : New sensors The system is able to work with new sensors technlogies growing on the market over the years.

3.9 Risk assessment

The system is confronted by several risks which are determined and mitigated in this section. Taking thoses risks into account permits to avoid them or at least reduce their impact.
The risk management involves : the identifcation of the risks, their probability and potential impact or consequences.

http://www.mitre.org/publications/systems-engineering-guide/acquisition-systems-engineering/risk-management/risk-impact-assessment-and-prioritization
Timeframe is classified in : Long , Medium , Short , Imminent
Consequences are classified in : Low, Moderate , High , Severe

Probability of Occurrence

1.00	Issue:	1	Certain to occur
0.95-0.99	High:	> 0.95 < 1	Extremely sure to occur
0.85-0.95	High:	> 0.85 <= 0.95	Almost sure to occur
0.75-0.85	High:	> 0.75 <=0.85	Very likely to occur
0.65-0.75	High:	> 0.65 <=0.75	Likely to occur
0.55-0.65	Medium:	> 0.55 <=0.65	Somewhat greater than an even chance
0.45-0.55	Medium:	> 0.45 <=0.55	An even chance to occur
0.35-0.45	Medium:	> 0.35 <=0.45	Somewhat less than an even chance
0.25-0.35	Low:	> 0.25 <=0.35	Not very likely to occur
0.15-0.25	Low:	> 0.15 <=0.25	Not likely to occur
0.00-0.15	Low:	> 0.00 <=0.15	Almost sure not to occur

3.9.1 Technical

T-RISK1	**The warning system does not detect floods in time**
Probability of occurrence	Low
Consequences	Severe because of the loss of human lives and money (estimate how much ?),loss of credibility regarding the end-users.
Prevention	Make sure the number of sensors is sufficient and that they are in good state (as low failure rate as possible : repair or replace them) thanks to a weekly checking for example.
Reaction	

T-RISK2	**The system sends warnings of a non-existing flood (false positive)**
Probability of occurrence	Low
Consequences	Moderate. People become more negligent to future messages.
Prevention	UAV's watching the area where the supposed flood is. Hiring an operator to control
Reaction	Send a message as soon as the mistake is detected to tell the population it was a false alert.

T-RISK3	**The system can't send messages to the necessary people because the communication platform is also destroyed by the flood.**
Probability of occurrence	Medium
Consequences	Severe . Loss of human lives and money
Prevention	Find another way to get the phone number of a population
Reaction	Using other medias : television, radio, Internet , ...

T-RISK4	**The system sends incorrect information, causing extra damage.**
Probability of occurrence	Low
Consequences	High . Loss of money and maybe lives.
Prevention	Operator checking the validity of the information sent by the system. Good collaboration with the insurance companies.
Reaction	

T-RISK5	**Hacker gets access to the system**
Probability of occurrence	Low
Consequences	High. The hackers may sent incorrect information deliberately during the flood : Loss of money and maybe human lives. The system isn't reliable anymore.
Prevention	Change password and hash codes every three months. Hire specialists in the security field.
Reaction	Update the security system / change it. Find a new algorithm for the creation of password and hash codes.

3.9.2 Business

B-RISK1	**Wrong estimation of the budget**
Probability of occurrence	Medium
Consequences	High . The final product hasn't the features expected.
Prevention	The team needs an accountant or at least someone taking care of the follow-up of the money.
Reaction	Remove some requirements or features of the product.

B-RISK2	**The money invested in the fabrication and achievement of the product/system is not covered by the sales (shortfall/deficit)**
Probability of occurrence	Medium
Consequences	High. Stopping the sale
Prevention	The team needs an accountant or at least someone taking care of the follow-up of the money.
Reaction	Adding more features to the product in order to make it more competitive in the market.

B-RISK3	**Competitors lowering their prices**
Probability of occurrence	Medium
Consequences	Moderate. Loss of money.
Prevention	
Reaction	

3.9.3 Schedule

S-RISK1	**The project is not finished at the deadline**
Probability of occurrence	Low
Consequences	Pressure for all the team members, loss of credibility regarding the end-users, selling a product with less features than expected.
Prevention	Meeting for the team members every week to keep track of the timing and take decisions according to the deadline.
Reaction	Remove some requirements or features in order to finish the project as soon as possible.

4 Analysis

This chapter describes the analysis. It lists the assumptions made about the system and it's environment. Next, an overview of the high-level design decisions is given.

4.1 Assumptions

There are several assumptions made about the system and it's environment:

1. The safety region / government has means to alert citizens in an area to evacuate.
2. The safety region is informed about our system and will alert citizens if needed when our system alerts the safety region.
3. People who are subscribed for flood warnings/guidance have a mobile phone that can receive text messages and MMS.
4. Sensors can be placed in the water ways and dikes in locations where they can provide representative measurements.
5. Information is available to the system with regards to the population density and terrain height in different areas.
6. A flood will not reach further than 50 km.
7. Supplying information about the areas affected by the flood in a resolution of 5 by 5 km (as described in FR-10) is sufficient for the government/safety region to make decisions regarding alerting and evacuating citizens.
8. It is possible to predict the expected water level up to 12 hours in the future.
9. A weather forecasting API is available that provides precipitation and wind data.
10. The sensors are (in some way) connected to the system.

I agree with group 2 that 10 is not really an assumption, but a functional requirement.

4.2 High-level Design Decisions

This section discusses the high-level design decisions.

Each decision is explained in a table. The arguments section of the table lists for each alternative a score for every important quality attribute. A higher score means a more favourable result. For example, a high score for costs means a low cost for the system.

Table 4.1: Decision – Operating system

Name	**Linux**
Decision	**DEC-1**
Status	**New**
Problem/Issue	The warning system software for the natural disasters need a platform to work on.
Decision	The warning system will use Linux as a platform. Based on Unix, Linux is a free platform that has proven itself and is used by many servers. It's open source meaning that everyone can check out how it works.
Alternatives	*Windows* Operating system is a closed platform developed by one of the biggest tech companies who provide a big development environment with it. *OpenBSD* A Unix-based system that is famous for it's proactive security and runs most of the Linux applications. However, some software packages aren't certified to run on OpenBSD, but are for Linux.
Arguments	

	Reliability	Resilience	Performance	Interopertability	Security	Scalability	Cost	**Score**
Weight	1	1	1	1	1	1	1	
Linux	4	4	4	4	4	4	5	29
Windows	3	2	3	2	3	3	1	17
OpenBSD	5	5	4	3	5	4	3	29

Table 4.2: Decision – Connectivity of the sensors

Name	**Connectivity of the sensors**
Decision	**DEC-2**
Status	**New**
Problem/Issue	The sensors need to deliver their data to the system and are located outdoors with at least 100m distance between each other.
Decision	The sensors will send their data to the system using mobile broadband. Using cellphone towers to communicate with the system.
Alternatives	*Landline* Connecting the sensors to the telephone network and use that network to communicate with the server.
Arguments	

	Reliability	Resilience	Performance	Interopertability	Security	Scalability	Cost	**Score**
Weight	1	1	1	1	1	1	1	
Mobile broadband	4	4	4	4	2	5	4	27
Landline	2	2	3	4	3	3	5	22
Satellite	3	1	2	4	4	4	3	21
Direct lines	2	1	5	2	5	1	1	17

Table 4.3: Decision – Cassandra Database

Name	**Cassandra Database**
Decision	**DEC-3**
Problem/Issue	A reliable database, which is the best in scalability and availability is needed to store our data for further processing and analysis.
Decision	Smart Flood Monitoring system will use Cassandra, which will run on top of the Linux platform, to store great amount of data from huge sensor arrays needed to carry out analytics and logging.
Alternatives	*Redis* Redis is a database that is best for storing data that changes rapidly with foreseeable database size which mostly fits in memory. This database is good to store real-time stock prices. *MongoDB* MongoDB is suitable for a database that needs dynamic queries. Indexes are mainly needed to runs this database system rather than Map/Reduce functions. *HBase* HBase is also written in Java. HBase is the database for Hadoop. This database is the best way to run Map/Reduce tasks on huge datasets.
Arguments	A short discussion of the cons and pro's is described by the table below

	Reliability	Resilience	Performance	Interoperability	Security	Scalability	Cost	**Score**
Weight	1	1	1	1	1	1	4	
Cassandra	4	3	5	3	3	5	4	27
Redis	2	3	3	3	3	2	4	20
MongoDB	2	3	3	3	3	3	4	21
HBase	3	3	4	3	3	4	4	24

I wrote this one myself. Please review this one yourself. :)

Table 4.4: Decision – Type of water level sensors

Name	**Type of water level sensor**
Decision	**DEC-4**
Problem/Issue	To measure the water level in the water ways and along the coast, a sensor is needed to measure the water level.
Decision	The system will use pressure sensors to measure the water level. Pressure sensors are submerged at a fixed level in the water body. By measuring the pressure on the sensor of the water above it, these types of sensors are able to determine the water level.
Alternatives	*Float-operated sensor* These types of sensors are mechanical and have a floating element which can move up and down with the water level. The floating element is protected in a 'stilling well' and therefore, the risk of damage is low. *Non-contact sensor* These types of sensors use (ultra)sonic waves to determine the water level. Sediment in the water can cause issues with the measurements. *Bubbler sensors* Bubbler sensors can measure the water level by measuring the pressure needed to force an air bubble through a submerged tube. Bubbler sensors have good resistance against damage from floods and debris.
Arguments	For this decision the interoperability and security have a weight of 0, because those depend on the specific sensor used and not on the type of sensor. Scalability was not taken into account because this depends mostly on the costs of the type of sensor.

	Reliability	Resilience	Performance	Interopertability	Security	Scalability	Cost	**Score**
Weight	3	2	1	0	0	0	3	
Pressure sensor	3	3	3	-	-	-	4	30
Float-operated sensor	3	4	3	-	-	-	3	29
Non-contact sensor	2	4	3	-	-	-	2	23
Bubbler sensors	4	5	2	-	-	-	1	29

Table 4.5: Decision – Cloud computing provider

Name	**Cloud service provider**
Decision	**DEC-5**
Problem/Issue	To connect all sensors and to process all the data, we need computing power.
Decision	The system will use Cloud VPS as the cloud provider. This provider is located in europe and can provide all services we need. This also means the privacy of the data is assured.
Alternatives	*Microsoft Azure* Microsoft Azure gives a good performance service. However the big disadvantage of this provider is that is not located in europe. This means different laws could apply to the data than in europe. *Amazon* The pricing and performance of Amazon is good. Unfortunately the same as for Azure also applies to Amazon. We can't be sure where our data is stored. *Own server implementation* Another option is to maintain our own server park. This will give us more freedom in terms of technical details. However, it will be way more expensive to maintain our own park.
Arguments	For this decision we will not take resilience into account. Resilience can also be seen as reliability and interoperability.

	Reliability	Resilience	Performance	Interopertability	Security	Scalability	Cost	**Score**
Weight	3	-	2	2	3	2	2	
Cloud VPS	4	-	4	3	4	5	3	54
Azure	4	-	4	3	3	5	3	51
Amazon	4	-	4	3	3	5	3	51
Own server park	4	-	4	5	4	3	1	50

5 System architecture

This chapter describes the general system architecture of the Smart Flood Monitoring. This will be described in these three sections: initial model, elaborated model, and verification.

5.1 Initial Model

5.2 Elaborated Model

5.3 Verification

6 Hardware Architecture

7 Software Architecture

This chapter describes the software architecture of the Smart Flood Monitoring. This will be described in these three sections: software architecture design (attribute-driven design), architecural view, and components.

7.1 Software architecture design

7.2 Architectural view

7.3 Components

8 Architecture evaluation

9 System evolution

A Time Tracking

A.1 Week 1

Person	Task	Hours
Eedema	Reviewing the document, reading the assignment, initializing requirements, & installing environment for project	8
Putra	Initial preparation for the course	5
Fakambi	Reading the document and assignment, Preparation and drafts with ideas	5
Schaefers	Setting up the working environment, create the context page and analysis page drafts. Setting up and improving the the document structure.	8
Klinkenberg		
Brandsma	Creating working environment, reading assignment, first draft business part	8
Menninga	Reading assignment, setting up working environment, first non-functional requirements	5

A.2 Week 2

Person	Task	Hours
Eedema	Coaching session, project planning session and work on business information chapters	9
Putra	Coaching session, project planning session, project meeting, first version of stakeholder part of requirements	7.5
Fakambi	Coaching session , project meeting, work on Non functional requirements	7
Schaefers	First coaching session, improved and enhanced the context and business information chapters. Also created a quality attributes prioritization table.	8
Klinkenberg	Coaching session, meetings, providing feedback on requirements	5.5
Brandsma	First version of use-cases, coaching session, meeting, use-cases, architectural vision	6.5
Menninga	First version of the functional requirements, coaching session, meeting	10.25

A.3 Week 3

Person	Task	Hours
Eedema	Coaching, meetings, analysis, business part, reviewing	14
Putra	Coaching session, meetings, proofread on chapter 1 and 2, revising stakeholders, database decision part of analysis, and preparing LaTeX file for the presentation	10
Fakambi	Coaching session, meetings, Non functional requirements and Risk assessment	10.5
Schaefers	Coaching session, meetings, reviewing, Business section	12
Klinkenberg	Coaching session, meetings, technical requirements, analysis, reviewing	13.5
Brandsma	Coaching session, meeting, architectural vision, use-cases, analysis	9
Menninga	Coaching session, meetings, updates functional requirements, reviewing entire document, updated assumptions and some improvements to structure of analysis, added decision about type of water level sensor.	14.0

A.4 Week 4

Person	Task	Hours
Eedema		
Putra		
Fakambi		
Schaefers		
Klinkenberg		
Brandsma	Coaching session, Meeting, Reviewing group 1, architectural vision, use-cases	10
Menninga	Coaching session, presentation prep., meeting, improvements FR and risks	10.5

A.5 Todo

Bibliography

[1] Dike monitoring and conditioning system (dmc). `http://www.dmc-system.com/`.

[2] A.L. Pyayt. *Combining Data-driven Methods with Finite Element Analysis for Flood Early Warning Systems.* PhD thesis, University of Amsterdam, 2015. http://hdl.handle.net/11245/1.478298.

[3] K. Wiegers and J. Beatty. *Software Requirements.* Developer Best Practices. Pearson Education, 2013.