



HACETTEPE UNIVERSITY

**BBM485**

**Software Architectures**

**Assignment 1**

**Fatma Çiğdem Tosun**

**216-----**

**1. Identify the context of the software. Please explain the main concepts of your context at maximum 1000 words.**

This software is to be executed in mobile phones, since it is a mobile messaging application.

The user must have a mobile phone number in order to login to the system, as login requires a mobile phone number.

Every profile has their own data, such as profile name, profile photo and status. The user can change their profile data, if they adhere the constraints listed below:

- Status must be at most 70 characters long.
- Profile photo must not be bigger than 250 MB.
- Profile name must be 10 characters or less.

The user can have a directory of friends for easier messaging. In order to add a friend to the directory of friends, user should use the number of the friend and should assign a name to the number.

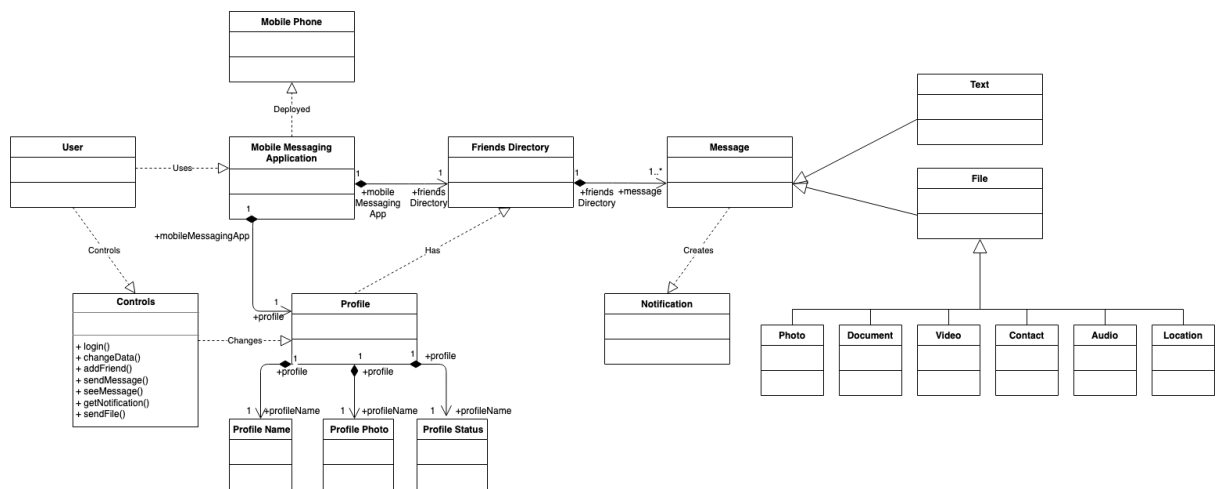
The user must be able to send messages. Messages can be sent to friends in the directory. The user also should be able to receive messages.

The user must be able to see the messages that have been sent or received as categorized according to the user whom have been communicated with, so the user should see the history of messaging.

The user must get notification when a message has been received. Notification should contain the name of the user whom send the message and the message that been sent. If the user clicks at the notification, the messaging application will be opened and the user will see the chat history with the user whom has sent the message.

Aside from sending messages, the user also must be able to sent files which are photos, documents, videos, audio,contact and location.

## 2. Model the context using appropriate modeling diagram.



### 3. Identify at least 5 concerns. Please explain the details of your concerns, including the importance to the stakeholders.

#### **Concern 1: Security**

Since this is a messaging application, security must be the number one concern because messages contain personal data and personal data must not be exposed to third-parties whom have not granted access by the owner of the personal data. Software Engineers must use best coding practices for security. Quality Assurance Engineer must check the given results from the Software Test Engineers whom must test for probable vulnerabilities and report them.

#### **Concern 2: Maintainability**

This application requires maintainability as there will be:

- bugs that need bug fixing which involves searching for errors and correcting them to allow the software to run seamlessly.
- capacity enhancement needs as enhancing the software will provide new features required by customers.
- replacement needs as replacing unwanted functionalities will improve adaptiveness and efficiency.
- security issues as fixing security vulnerabilities found in your proprietary code or third-party code, especially open source components.

Software Engineers should consider maintainability while coding. Quality Assurance Engineer must check whether the product covers maintainability quality concerns.

#### **Concern 3: Responsiveness**

Since messaging is real-time and the people that message expects messaging to be like a call, mobile messaging application should have the ability to meet its objectives for response time or throughput.

Software Test Engineers should test all of the functions to ensure that they are fast enough. Software Engineers should optimize their code to make the system as fast as possible. Quality Assurance Engineer must check whether the product covers responsiveness quality concerns.

#### **Concern 4: Learnability**

In 2020, messaging is quite the essential need of a human, so the mobile messaging application will be used by almost everybody. This means that, it should be easily understandable. Learnability will allow users to quickly become familiar with the product to make good use of all of the product's features and capabilities.

Software Test Engineers should conduct tests on user experience and ensure that the application can be learned quickly. Software Engineers should consider learnability while planning and coding. Quality Assurance Engineer must check whether the product covers learnability quality concerns.

### **Concern 5: Scalability**

High traffic is expected from the mobile messaging application, so the system should be able to handle a growing amount of work by adding resources to the system. Software Development Manager and Product Owner must ensure that the system has enough resources to support increasing number of users. Quality Assurance Engineer should check whether the product covers scalability quality concerns.

### 4. Choose your views, and visualize the importance for stakeholders.

#### Process View

Process view is used to explain the system processes and how they communicate. Process view focuses on the run time behavior of the system. The process view addresses scalability and responsiveness.

#### Development View

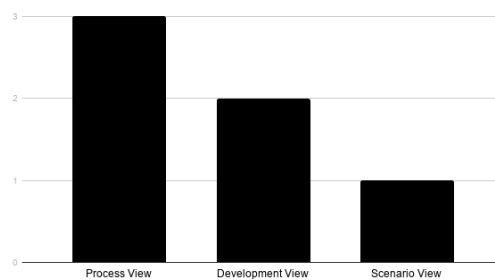
Development view is used to illustrate a system from programmer's perspective. Development view is concerned with software management. This view covers maintainability and security concerns

#### Scenario View

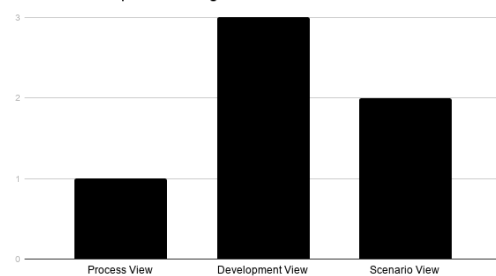
Scenario view, also known as use case view, is used to identify architectural elements and to illustrate and validate the architectural design. This scenario view addresses learnability concerns.

#### The importance of views for stakeholders:

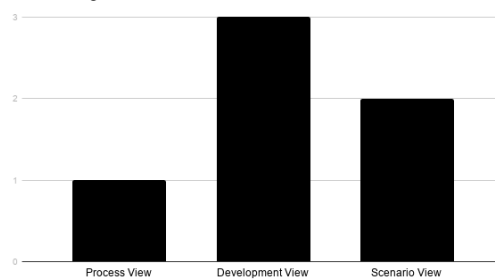
Product Owner



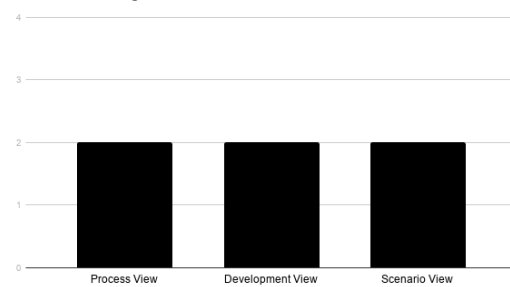
Software Development Manager



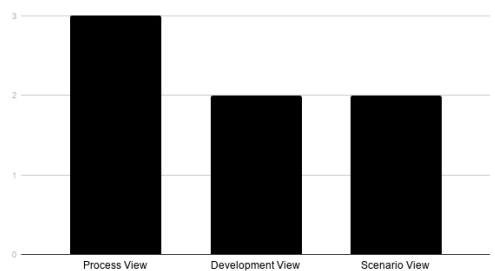
Software Engineer



Software Test Engineer



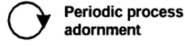





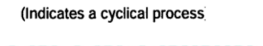


Quality Assurance Engineer









## 5. Define viewpoints. Please use Eclipse Sirius to create the viewpoint.

## Process Viewpoint

<b>Name</b>	Process
<b>Stakeholders</b>	Product Owner, Quality Assurance Engineer
<b>Concerns</b>	Scalability, Responsiveness
<b>Elements</b>	Process - An instance to host or process the system.
<b>Relations</b>	The relationship for which data should go to which process.
<b>Constraints</b>	Cost
<b>Notation</b>	<div> <div> <b>Components</b>  <p>Process</p>  <p>Simplified Process</p>  <p>Periodic process adornment</p> </div> <div> <b>Connectors</b>  <p>Unspecified</p>  <p>Message</p>  <p>Remote Procedure Call</p>  <p>Message, bidirectional</p>  <p>Event broadcast</p> <p>(Indicates a cyclical process)</p>  </div> </div>

### Development Viewpoint

<b>Name</b>	Development
<b>Stakeholders</b>	Software Development Manager, Software Engineer
<b>Concerns</b>	Maintainability, Security
<b>Elements</b>	Module - Software Modules Subsystem Layer
<b>Relations</b>	Reference Compilation Dependency
<b>Constraints</b>	Layers must be ordered.
<b>Notation</b>	<div> <div> <b>Components</b>   <b>Module</b>   <b>Subsystem</b>   <b>Layer</b>   </div> <div> <b>Connectors</b>   <b>Reference Compilation dependency (include, "with")</b>   </div> </div>



**Scenario Viewpoint**

<b>Name</b>	Scenario
<b>Stakeholders</b>	End-User, Software Developer
<b>Concerns</b>	Learnability
<b>Elements</b>	Step Script
<b>Relations</b>	
<b>Constraints</b>	
<b>Notation</b>	

6. **Model views using the viewpoints. Please use Eclipse Sirius to create your views. You are responsible to send the project files. Please add a instruction document to import project in pdf format.**

-

7. **Document architecture using RUP template. The document format shall be pdf. You can export your Eclipse Sirius diagrams as image files into your documentation.**

You can find the Software Architecture Document attached.