



# Bilkent University

**CS 415  
Software Product Line  
Engineering Project  
-Phase 2-**

Group: ProPic

Group Members:

Muhammed Çavuşoğlu  
Duygu Durmuş  
Özgür Can Erdoğan  
Mithat Orhan  
Arif Can Terzioğlu

<a href="mailto:m.cavusoglu@ug.bilkent.edu.tr">m.cavusoglu@ug.bilkent.edu.tr</a>	21400653
<a href="mailto:duygu.durmus@ug.bilkent.edu.tr">duygu.durmus@ug.bilkent.edu.tr</a>	21401178
<a href="mailto:ozgur.erdogan@ug.bilkent.edu.tr">ozgur.erdogan@ug.bilkent.edu.tr</a>	21300586
<a href="mailto:mithat@ug.bilkent.edu.tr">mithat@ug.bilkent.edu.tr</a>	21000792
<a href="mailto:arif.terzioglu@ug.bilkent.edu.tr">arif.terzioglu@ug.bilkent.edu.tr</a>	21302061

Instructor: Eray Tüzün

## **Table of Contents**

<b>1. Introduction</b>	<b>4</b>
<b>2. Requirements</b>	<b>4</b>
<b>3. Kano Methodology</b>	<b>5</b>
<b>4. Variants</b>	<b>7</b>
4.1 Personal Use	7
4.2 Professional Use	7
4.3 Enterprise Use	7
<b>5. Features</b>	<b>7</b>
5.1 Common Features	7
5.2 Variable Features	7
5.3 Product Specific Features	8
<b>6. Feature Model</b>	<b>8</b>
<b>7. Variability Configuration</b>	<b>9</b>
7.1 Personal Use	9
7.2 Professional Use	9
7.3 Enterprise Use	9
<b>8. High Level Architecture</b>	<b>10</b>
8.1 Licensing Use Case Model	10
8.2 Logo Use Case Model	13
8.3 Sharing Use Case Model	14
8.4 Customer Support Use Case Model	16
8.5 Upload Image Use Case Model	17
8.6 File Management Use Case Model	18
8.7 Pricing Options Use Case Model	19
<b>9. Prototype</b>	<b>20</b>
9.1 Software Reuse in Prototype	28
<b>10. Discussion</b>	<b>29</b>
10.1 Requirements	29
10.2 Feature Model	29
10.3 Kano Methodology	30
10.4 Orthogonal Model	30
10.5 Prototype	31
<b>11. Group Member Contribution</b>	<b>31</b>

## Abstract

LicenceMe is a photography licensing application with three different tiers: personal, professional and enterprise. Kano Methodology is used to identify requirements. Feature models of the application are described. High level architecture is illustrated by use case diagrams and orthogonal models. Implementation process and prototype are explained in detail.

**Keywords:** Photography, license, watermark, Kano Model, Feature Model, Orthogonal Model, Software Product Line Engineering.

## 1. Introduction

LicenceMe is a photography licensing application with three different tiers: personal, professional and enterprise with different type of users in mind. Users will have access to professional or enterprise tiers after an upgrade to their current tiers by a payment. Personal tier is created for hobbyists and it provides simple functionalities for licensing. Professional tier offers more features and it is created for personal professionals and small studios. Enterprise edition tier contains extended features with searchable archive and emergency hotline, and it is created for large media companies.

## 2. Requirements

Our initial meetings lead us to prepare a general set of requirements. Initial set included basic functionalities that enables us to create a minimum viable product. These basic functionalities are licensing, watermarking, customer support, sharing options and payment. We expanded this set to include any additional features that may increase customer satisfaction.

This expanded set includes archiving, searching, emergency hotline, professional share that facilitate sharing easily with printing house, graphic design consultants or internal design team.

## 3. Kano Methodology

Kano Model is developed in order to classify the customer requirements in the software. It is possible to identify the variability and satisfy customer demands beforehand. The answers of questionnaire enable to obtain Kano classification for requirements which can be converted into SPLE classification for defining variability. Consequently, prepared questionnaire for our project was answered by 5 group members so as to ensure that classification of customer needs is correct. The results of the Kano Model is given below:

Questionnaire	Like	Expected	Don't care	Can live with it	Dislike	Kano Classification	SPLE Classification
Suppose LicenseMe has social media support for photo sharing, what would you think about that?	xxxx	x				Basic Requirement	Core Asset
Suppose LicenseMe has no social media support for photo sharing, what would you think about that?					xxxxx		
Suppose LicenseMe enables photo sharing on Facebook, what would you think about that?	xxx	x	x			Satisfier	Variable
Suppose LicenseMe does not enable photo sharing on Facebook, what would you think about that?				xx	xxx		
Suppose LicenseMe enables photo sharing on Instagram, what would you think about that?	xxxxx					Satisfier	Variable
Suppose LicenseMe does not enable photo sharing on Instagram, what would you think about that?					xxxxx		
Suppose LicenseMe enables photo sharing on Twitter, what would you think about that?	xx	x	x	x		Satisfier	Variable
Suppose LicenseMe does not enable photo sharing on Twitter, what would you think about that?				x	xxx		
Suppose LicenseMe enables saving photos to camera roll, what would you think about that?	xxxx	x				Satisfier	Variable
Suppose LicenseMe does not enable saving photos to camera roll, what would you think about that?				x	xxxx		
Suppose LicenseMe has support for professional photo sharing, what would you think about that?	x	xxx	x			Basic Requirement	Core Asset
Suppose LicenseMe has no support for professional photo sharing, what would you think about that?				xx	xxx		
Suppose LicenseMe benefits from google drive while professional photo sharing, what would you think about that?	x		xxxx			Indifferent Requirement	Internal Variable
Suppose LicenseMe does not benefit from google drive while professional photo sharing, what would you think about that?		x	xxx	x			
Suppose LicenseMe enables professional sharing of photos with external design consultant, what would you think about that?	xxx	x	x			Delighter	Variable
Suppose LicenseMe does not enable professional sharing of photos with external design consultant, what would you think about that?			xx	xxx			
Suppose LicenseMe enables professional sharing of photos with internal design team, what would you think about that?	xx	x	x	x		Delighter	Variable
Suppose LicenseMe does not enable professional sharing of photos with internal design team, what would you think about that?			x	xxx	x		
Suppose LicenseMe enables professional sharing of photos with printing house, what would you think about that?	xxxx	x				Delighter	Variable
Suppose LicenseMe does not enable professional sharing of photos with printing house, what would you think about that?		xxxx	x				
Suppose LicenseMe provides licensing for photos, what would you think about that?		xxxxx				Basic Requirement	Core Asset
Suppose LicenseMe does not provide licensing for photos, what would you think about that?					xxxxx		

Figure 1: The figure showing the first part of kano questionnaire and classification results

Questionnaire	Like	Expected	Don't care	Can live with it	Dislike	Kano Classification	SPLE Classification
Suppose LicenseMe uses watermark of logos for licensing, what would you about that?		xxxx	x				
Suppose LicenseMe does not use watermark of logos for licensing, what would you about that?			x		xxxx	Basic Requirement	Core Asset
Suppose LicenseMe utilizes logo of "LicenseMe" as watermark for licensing, what would you think about that?	xxx		xx			Satisfier	Variable
Suppose LicenseMe does not utilize logo of "LicenseMe" as watermark for licensing, what would you think about that?				xx	xxx		
Suppose LicenseMe utilizes given logo by user as watermark for licensing, what would you think about that?	xxxx	x				Delighter	Variable
Suppose LicenseMe does not utilize given logo by user as watermark for licensing, what would you think about that?			xxxx	x			
Suppose LicenseMe provides positioning for the watermarks of logos, what would you think about that?	x	xxxx				Basic Requirement	Core Asset
Suppose LicenseMe does not provide positioning for the watermarks of logos, what would you think about that?				xx	xxx		
Suppose LicenseMe provides fixed positioning for watermarks of logos, what would you think about that?	xxx	xx				Satisfier	Variable
Suppose LicenseMe provides does not provide fixed positioning for watermarks of logos, what would you think about that?		x	x		xxx		
Suppose LicenseMe provides provides flexible positioning for watermarks of logos, what would you think about that?	xxxx	x				Delighter	Variable
Suppose LicenseMe provides does not provide flexible positioning for watermarks of logos, what would you think about that?		x	xxx	x			
Suppose LicenseMe offers payment methods for pricing, what would you think about that?		xx	xxx			Basic Requirement	Core Asset
Suppose LicenseMe does not offer payment methods for pricing, what would you think about that?			x	x	xxx		
Suppose LicenseMe offers PayPal as a payment method, what would you think about that?	xxx	x	x			Satisfier	Variable
Suppose LicenseMe does not offer PayPal as a payment method, what would you think about that?				x	xxx		
Suppose LicenseMe offers Apple Pay as a payment method, what would you think about that?	xx	x	x			Satisfier	Variable
Suppose LicenseMe does not offer Apple Pay as a payment method, what would you think about that?			x	x	xx		
Suppose LicenseMe offers credit card as a payment method, what would you think about that?	xxxx	x				Satisfier	Variable
Suppose LicenseMe does not offer credit card as a payment method, what would you think about that?				xxxx			

Figure 2: The figure showing the second part of kano questionnaire and classification results

Questionnaire	Like	Expected	Don't care	Can live with it	Dislike	Kano Classification	SPLE Classification
Suppose LicenseMe has offered plans for pricing, what would you think about that?		xxxx	x				
Suppose LicenseMe has no offered plans for pricing, what would you think about that?			x	x	xx	Basic Requirement	Core Asset
Suppose LicenseMe serves "Free" option as an offered plans for pricing, what would you think about that?	xxxxx					Satisfier	Variable
Suppose LicenseMe does not serve "Free" option as an offered plans for pricing, what would you think about that?					xxxxx		
Suppose LicenseMe serves subscription option as an offered plans for pricing, what would you think about that?	xxx	xx				Satisfier	Variable
Suppose LicenseMe does not serve subscription option as an offered plans for pricing, what would you think about that?			x	x	xx		
Suppose LicenseMe serves annual subscription option as an offered plans for pricing, what would you think about that?	xxx	x	x			Satisfier	Variable
Suppose LicenseMe does not serve annual subscription option as an offered plans for pricing, what would you think about that?		x		x	xx		
Suppose LicenseMe serves monthly subscription option as an offered plans for pricing, what would you think about that?	xxx		x			Satisfier	Variable
Suppose LicenseMe does not serve monthly subscription option as an offered plans for pricing, what would you think about that?			x		xxx		
Suppose LicenseMe serves customized price option as an offered plans for pricing, what would you think about that?	xx	x	x	x		Delighter	Variable
Suppose LicenseMe does not serve customized price option as an offered plans for pricing, what would you think about that?			xxx	x			
Suppose LicenseMe offers extension addition to offered plans, what would you think about that?	xxxxx					Satisfier	Variable
Suppose LicenseMe does not offer extension addition to offered plans, what would you think about that?			xx		xxx		
Suppose LicenseMe has license choices, what would you think about that?		xxxxx				Basic Requirement	Core Asset
Suppose LicenseMe has no license choices, what would you think about that?				x	xxx		
Suppose LicenseMe enables one license choice only, what would you think about that?	xxx	x	x			Satisfier	Variable
Suppose LicenseMe does not enable one license choice only, what would you think about that?				x	xxx		
Suppose LicenseMe enables limited license choice, what would you think about that?	xxx		x			Satisfier	Variable
Suppose LicenseMe does not enable limited license choice, what would you think about that?			x	x	xxx		
Suppose LicenseMe enables unlimited license choice, what would you think about that?	xxx		x			Delighter	Variable
Suppose LicenseMe does not enable unlimited license choice, what would you think about that?			xxx	x			

Figure 3: The figure showing the third part of kano questionnaire and classification results

Questionnaire	Like	Expected	Don't care	Can live with it	Dislike	Kano Classification	SPLE Classification
Suppose LicenseMe has license types, what would you think about that?			xxx	xx			
Suppose LicenseMe has no license types, what would you think about that?			x		xxx	Basic Requirement	Core Asset
Suppose LicenseMe offers CC as a license type, what would you think about that?	xxx	x	x			Satisfier	Variable
Suppose LicenseMe does not offer CC as a license type, what would you think about that?		x	x	x	xx		
Suppose LicenseMe offers RF as a license type, what would you think about that?	xxx		x			Satisfier	Variable
Suppose LicenseMe does not offer RF as a license type, what would you think about that?				xx	xxx		
Suppose LicenseMe offers RM as a license type, what would you think about that?	xxx	xx				Delighter	Variable
Suppose LicenseMe does not offer RM as a license type, what would you think about that?			xxxxx	x			
Suppose LicenseMe offers micro as a license type, what would you think about that?	xxx		xx			Undesired Requirement	Not Added
Suppose LicenseMe does not offer micro as a license type, what would you think about that?	xxx	x	x				
Suppose LicenseMe offers subscription as a license type, what would you think about that?			x	xxx		Undesired Requirement	Not Added
Suppose LicenseMe does not offer subscription as a license type, what would you think about that?	xx	x	x				
Suppose LicenseMe enables user to upload photos for licensing, what would you think about that?		xxxxx				Basic Requirement	Core Asset
Suppose LicenseMe does not enable user to upload photos for licensing, what would you think about that?				xxxxx			
Suppose LicenseMe accepts uploading photos with different image qualities, what would you think about that?	xxx	x	x			Basic Requirement	Core Asset
Suppose LicenseMe does not accept uploading photos with different image qualities, what would you think about that?			x		xxx		
Suppose LicenseMe accept low image quality for uploaded photos, what would you think about that?	xxxx	x				Satisfier	Variable
Suppose LicenseMe does not accept low image quality for uploaded photos, what would you think about that?		x	x		xxx		
Suppose LicenseMe accept medium image quality for uploaded photos, what would you think about that?	xxx	xx				Satisfier	Variable
Suppose LicenseMe does not accept medium image quality for uploaded photos, what would you think about that?			x		xxx		
Suppose LicenseMe accept high image quality for uploaded photos, what would you think about that?	xxx	x	x			Delighter	Variable
Suppose LicenseMe does not accept high image quality for uploaded photos, what would you think about that?		x	xxx				

Figure 4: The figure showing the fourth part of kano questionnaire and classification results

Questionnaire	Like	Expected	Don't care	Can live with it	Dislike	Kano Classification	SPLE Classification
Suppose LicenseMe enables naming photos after licensing, what would you think about that?	***	**				Basic Requirement	Core Asset
Suppose LicenseMe does not enable naming photos after licensing, what would you think about that?				****			
Suppose LicenseMe offers file managing for photos, what would you think about that?	***	**				Delighter	Variable
Suppose LicenseMe does not offer file managing for photos, what would you think about that?	*	***	*				
Suppose LicenseMe enables searching photos, what would you think about that?	***	**				Satisfier	Variable
Suppose LicenseMe does not enable searching photos, what would you think about that?			*	*	***		
Suppose LicenseMe uses binary search algorithm for searching, what would you think about that?	****	*				Indifferent Requirement	Internal Variable
Suppose LicenseMe does not use binary search algorithm for searching, what would you think about that?	*	***					
Suppose LicenseMe offers local archive for licensed photos, what would you think about that?	****	*				Delighter	Variable
Suppose LicenseMe does not offer local archive for licensed photos, what would you think about that?	*	*	***				
Suppose LicenseMe offers grouping for licensed photos, what would you think about that?	***	*				Delighter	Variable
Suppose LicenseMe does not offer grouping for licensed photos, what would you think about that?	*	****					
Suppose LicenseMe offers uploading 50 photos per month as an upload limit, what would you think about that?						Undesired Requirement	Not Added
Suppose LicenseMe does not offer uploading 50 photos per month as an upload limit, what would you think about that?							
Suppose LicenseMe offers uploading 100 photos per month as an upload limit, what would you think about that?	***	**				Satisfier	Variable
Suppose LicenseMe does not offer uploading 100 photos per month as an upload limit, what would you think about that?			*	***			
Suppose LicenseMe offers uploading 1000 photos per month as an upload limit, what would you think about that?	****	*				Satisfier	Variable
Suppose LicenseMe does not offer uploading 1000 photos per month as an upload limit, what would you think about that?	*	**	***				
Suppose LicenseMe offers uploading unlimited photos, what would you think about that?	****	*				Delighter	Variable
Suppose LicenseMe does not offer uploading unlimited photos, what would you think about that?	*	****					
Suppose LicenseMe offers customer support, what would you think about that?	*****					Basic Requirement	Core Asset
Suppose LicenseMe does not offer customer support, what would you think about that?		*	***				
Suppose LicenseMe offers daytime customer support, what would you think about that?	*	***	*			Basic Requirement	Core Asset
Suppose LicenseMe does not offer daytime customer support, what would you think about that?		*	*	***			
Suppose LicenseMe offers emergency hotline as customer support option, what would you think about that?	****	*				Delighter	Variable
Suppose LicenseMe does not offer emergency hotline as customer support option, what would you think about that?	*	***	*				
Suppose LicenseMe supports pop-up advertisements, what would you think about that?			*	***		Undesired Requirement	Not Added
Suppose LicenseMe does not support pop-up advertisements, what would you think about that?	*****						

Figure 5: The figure showing the last part of kano questionnaire and classification results

## 4. Variants

### 4.1 Personal Use

User is interested in photography as hobby. She can licence 100 photographs per month for free. This tier watermarks photographs with “Licence Me” logo and watermark has a fixed position. There is no option for selecting licence type. Only Creative Commons licence (photographs can be used but link must be given) is available. The customer is able to share photographs to Facebook, Twitter and other social media services, save them to camera roll and name them. The customer support is available.

### 4.2 Professional Use

User is interested in photography professionally such as photographers and small studios. She can licence 1000 photographs per month. User pays \$50 per month (billed annually) or \$75 per month and opt-out anytime. If user exceeds upload limit, she can pay \$10 or \$25 for licensing 100 or 500 photographs respectively. This tier watermarks photographs with Licence Me logo, additionally user’s own logo and fixed position for watermark is applied. There are limited options for selecting licence type. The customer is able to share photographs to Facebook, Twitter and other social media services, save them to camera roll and name them. The customer support is available.

### 4.3 Enterprise Use

Users of this tier are media companies and big studios. The company can licence unlimited number of photographs. A custom price is determined for this tier. This tier watermarks photographs with customer’s own logo and the position of watermark can be selected. There are unlimited options for selecting licence type. The customer is able to share photographs to Facebook, Twitter and other social media services, save them to camera roll and name them. The photographs can be saved to the application locally. There is a searchable archive and filing feature. Customer support and emergency hotline are available. If it is desired, users

will be able to send their works to printing house, graphics designers, interior designers in their preferences.

## 5. Features

### 5.1 Common Features

- **Share Option:** User can share photographs to social media and save them to Camera Roll.
- **Naming Photographs:** User can name licensed image.
- **Uploading & Licensing Photographs:** This is the core feature of LicenseMe.
- **Customer Support:** Users can call customer support using call button.

### 5.2 Variable Features

- **Watermark Style:** In personal tier, users can only use LicenseMe Logo. In professional and enterprise tiers, users can use a custom logo for watermarking.
- **Watermark Position:** In personal and professional tiers, users cannot select position of the watermark. In enterprise tier, users can select position of the watermark.
- **Licence Choice:** In personal tier, users can only use Creative Commons license. In professional tier there is a limited set of licenses. In enterprise tier, license choice is not limited.
- **Pricing:** Personal tier is free to use. Pricing for professional tier is based on a subscription model. A custom price is determined for each enterprise tier use.
- **Upload Limit:** In personal tier, users can license 100 photos per month, in Professional tier, users can license 1000 photos per month, and in enterprise tier, there is no upload limit.
- **Image Quality:** Quality of licensed image is Low, Medium, High for Personal, Professional, Enterprise tiers respectively.
- **Payment Method:** Three payment methods are accepted (PayPal, Credit Card, Apple Pay). Users in professional or enterprise tiers can pay via Apple Pay.

### 5.3 Product Specific Features

- **Local Archive of Photographs:** A local archive of licensed photographs is available to users in enterprise tier.
- **Searching Photographs:** Users in enterprise tier can search photos in the local archive.
- **Grouping:** Enterprise users can create directories and group photographs.
- **Emergency Hotline:** An additional customer support service which is available 24/7 is provided to enterprise users.
- **Professional Share:** After licensing the photographs, users can send their works easily to a printing house, graphics designer or internal design team.

## 6. Feature Model

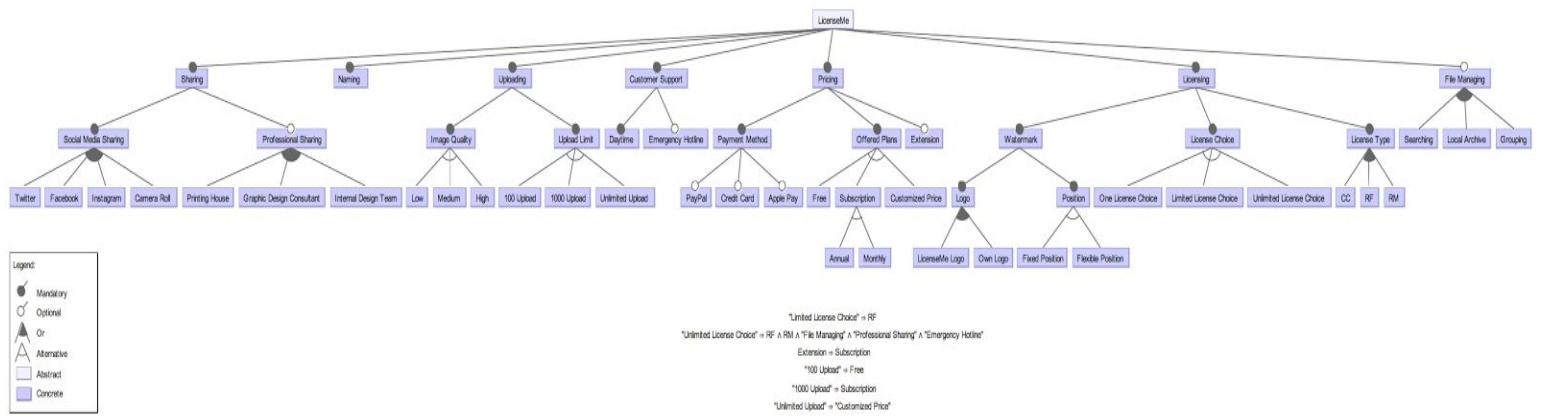


Figure 6: The figure showing the feature model of LicenseMe

Number of configurations: more than 225971

Number of program variants: more than 226113

Figure 7: The figure showing number of configurations and program variants of feature model calculated with eclipse

## 7. Variability Configuration

### 7.1 Personal Use

For personal tier, users can benefit from several features. These features are marked with red dots in feature model and given below:

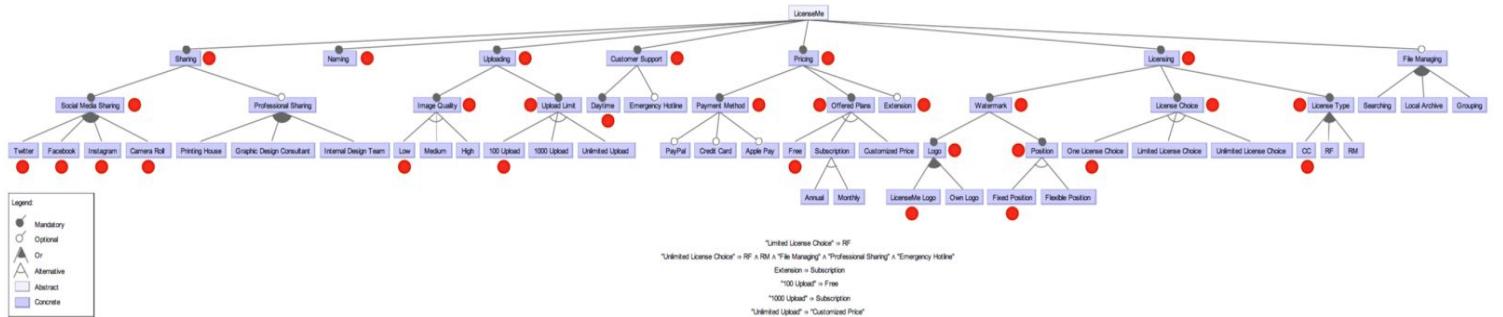


Figure 8: The figure showing capabilities of personal tier

## 7.2 Professional Use

For professional tier, users can benefit from several features. These features are marked with green dots in feature model and given below:

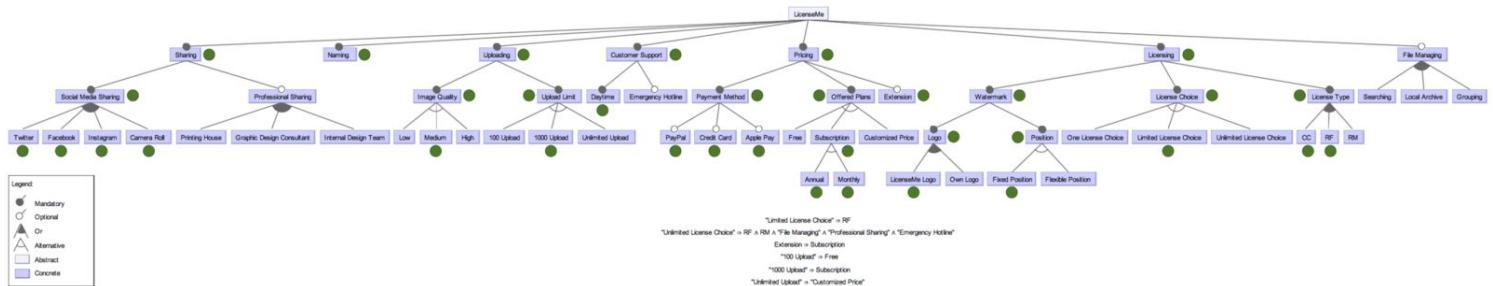


Figure 9: The figure showing capabilities of professional tier

## 7.3 Enterprise Use

For enterprise tier, users can benefit from several features. These features are marked with purple dots in feature model and given below:

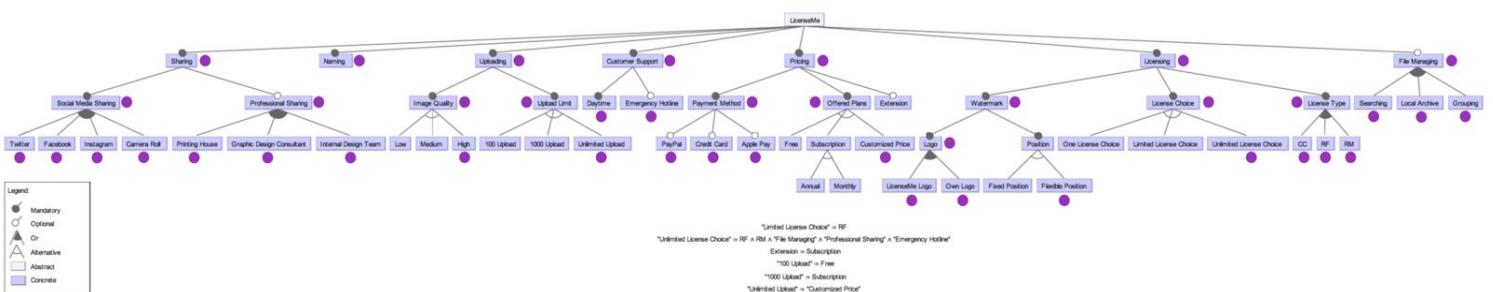


Figure 10: The figure showing capabilities of enterprise tier

## 8. High Level Architecture

### 8.1 Licensing Use Case Model

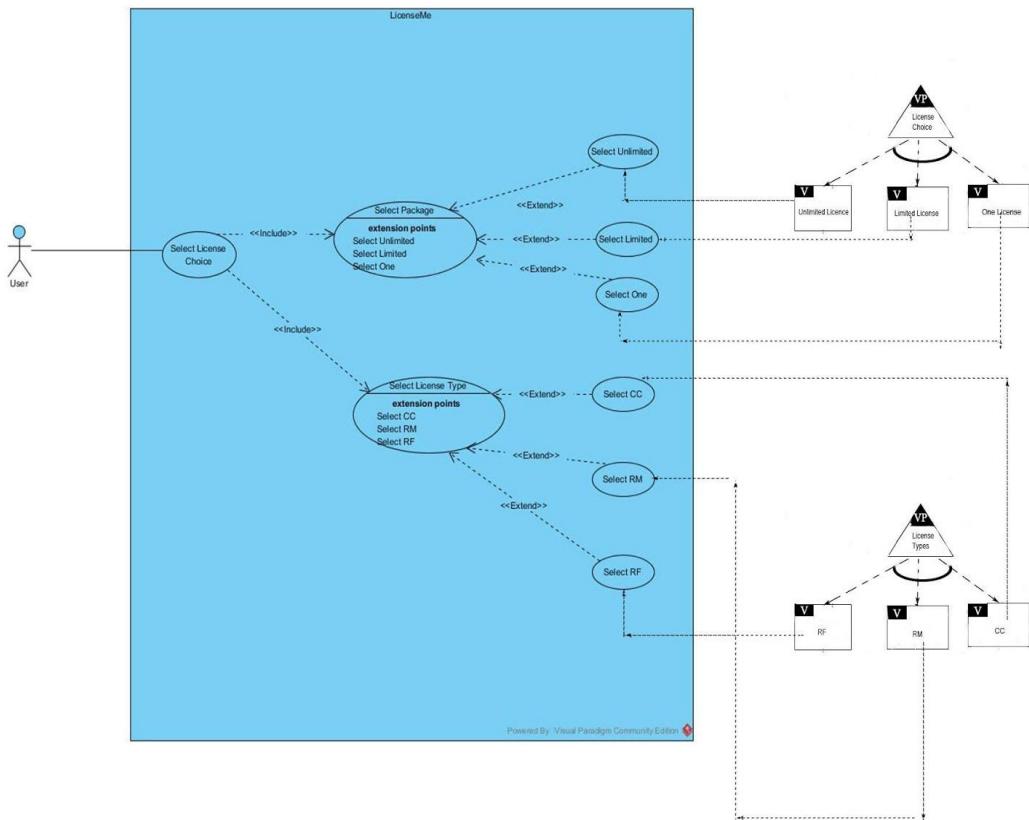


Figure 11: The figure showing the use case model for licensing

This model describes the selection of license packaging and license type. User can choose three types of license to apply on her/his own photos. Users are limited by package type which they select. A personal user has one license choice which means that he/she can only select CC as a license type, professional user has limited license choice so he/she can select among CC and RF license types while enterprise user has unlimited license choice which enables user to select any license types offered.

Name	User selects License Type
Primary Actor	User
Preconditions	User should select any plan
Main Scenario	<ol style="list-style-type: none"> <li>1. User enters the "LicenseMe" application and comes to select plan screen.</li> <li>2. User selects Personal use packet which is free (One license right is assigned to account).</li> <li>3. From the License type slider container, user selects creative commons type.</li> <li>4. User clicks the camera symbol and selects the photograph which will be watermarked.</li> <li>5. User gets the watermarked image with Creative Common License type.</li> </ol>
Extensions	<p>*User can change her/his plan type to Professional Use.</p> <ol style="list-style-type: none"> <li>2. User selects Professional use packet which is free (Limited right is assigned to account).</li> <li>3. From the License type slider container, user selects one of Creative Common and Royalty Free License.             <ol style="list-style-type: none"> <li>3a. User selects Creative Common.</li> <li>3b. User selects Royalty Free.</li> </ol> </li> <li>4. User clicks the camera symbol and selects the photograph which will be watermarked.</li> <li>5. User gets the watermarked image with Creative Common or Royalty Free License type.</li> </ol> <p>*User can change her/his plan type to Enterprise Use.</p> <ol style="list-style-type: none"> <li>2. User selects Enterprise use packet which is free (Unlimited right is assigned to account).</li> <li>3. From the License type slider container, user selects Creative Commons or Royalty Free or Rights Managed type.             <ol style="list-style-type: none"> <li>3a. User selects Creative Common.</li> <li>3b. User selects Royalty Free.</li> <li>3c. User selects Rights Managed.</li> </ol> </li> <li>4. User clicks the camera symbol and selects the photograph which will be watermarked.</li> <li>5. User gets the watermarked image with one of Creative Common and Royalty Free License and Rights Managed type.</li> </ol>

## 8.2 Logo Use Case Model

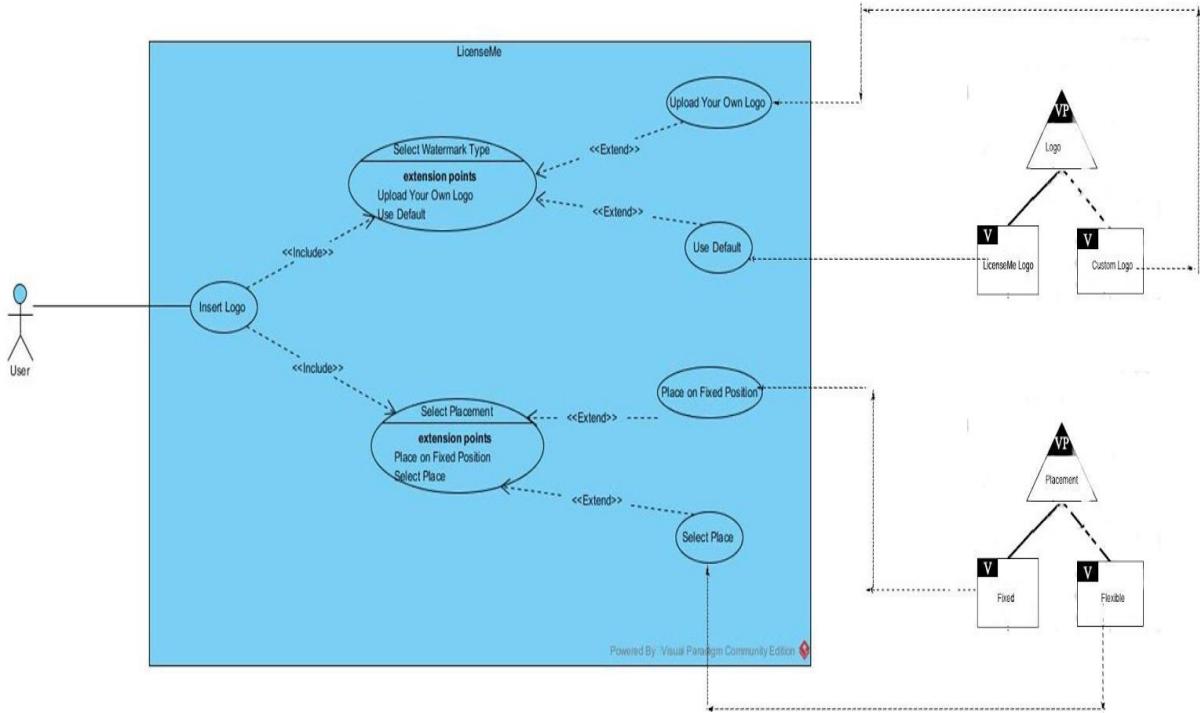


Figure 12: The figure showing the use case model for logo selection and positioning

Users can watermark their photos with default logo of LicenseMe or customized logo they upload. The uploaded photos can be licensed with LicenseMe logo for personal while enterprise and professional users can license their photos with their own logo. Similarly, the fixed watermark position is available for all users while flexible watermark position is only available for only enterprise users They can place the watermark either on default or desired position.

Name	User selects Logo Type
Primary Actor	User
Preconditions	User should select any plan and select any image as a watermark.
Main Scenario	<ol style="list-style-type: none"> <li>1. User enters the “LicenseMe” application and comes to select plan screen.</li> <li>2. User selects Personal use packet which is free.</li> <li>3. User clicks the camera symbol and selects the photograph which will be watermarked.</li> <li>4. User gets the initial watermarked image that “LicenseMe” offers at fixed position.</li> </ol>
Extensions	<p>*User can change her/his plan type to Professional Use or Enterprise Use.</p> <ol style="list-style-type: none"> <li>2. User selects Professional or Enterprise use packet.</li> <li>3. User clicks the camera symbol and selects the photograph which will be watermarked.</li> <li>4. User gets the customized watermarked image at any flexible position.</li> </ol>

### 8.3 Sharing Use Case Model

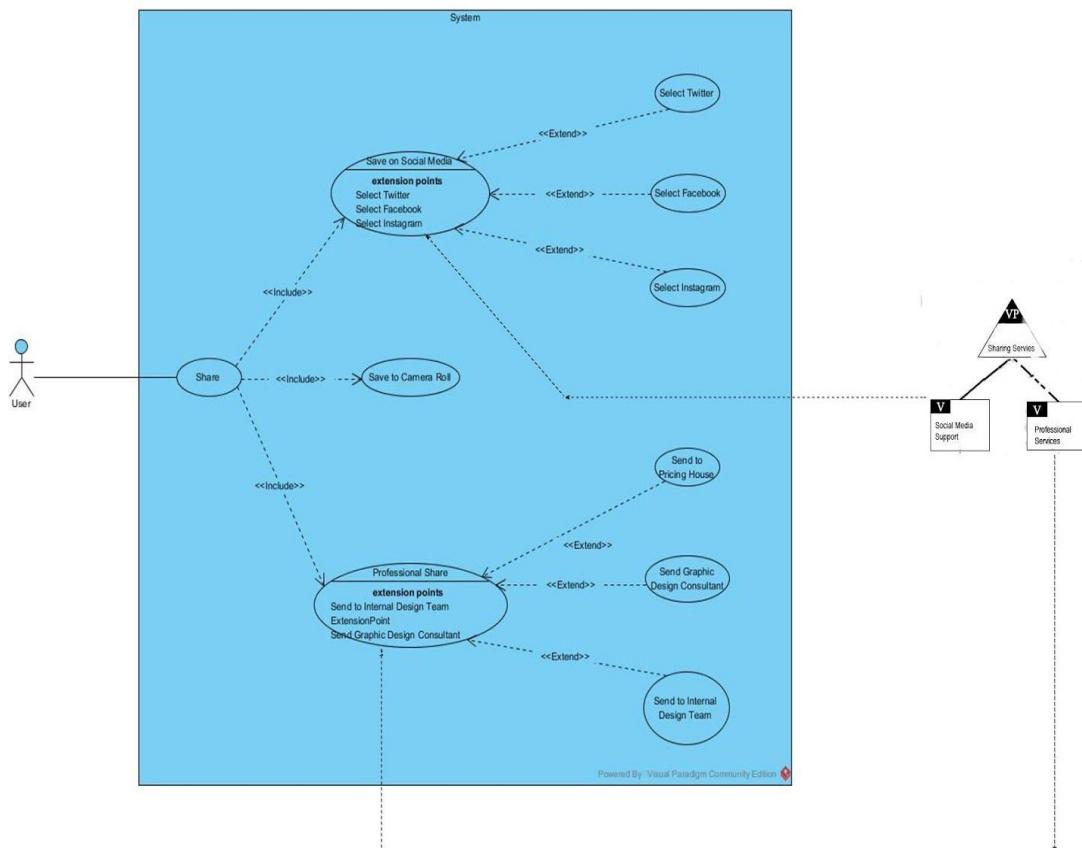


Figure 13: The figure showing the use case model for sharing licensed photos

Users can share their watermarked images in three ways which are professional share, saving to camera roll and social media. Sharing on Twitter, Facebook and Instagram are provided within the system. If Enterprise usage is valid, professional sharing enables users to sharing sending their watermarked images to pricing house, graphic design consultant and internal design team.

Name	User selects Sharing Services
Primary Actor	User
Preconditions	User should select any plan and click to share button.
Main Scenario	<ol style="list-style-type: none"> <li>1. User enters the “LicenseMe” application and comes to select plan screen.</li> <li>2. User selects Personal or Professional use packet or change plan to Professional use from Personal use.</li> <li>3. User clicks the share button.</li> <li>4. User shares its image on social media selected.</li> </ol>
Extensions	<p>*User can change her/his plan type to Enterprise Use and share via professional services.</p> <ol style="list-style-type: none"> <li>2. User selects Enterprise use packet or change plan to Enterprise use from other use.</li> <li>3. User clicks the professional share button.           <ol style="list-style-type: none"> <li>3a. User selects social share.</li> <li>3b. User selects professional share.</li> </ol> </li> <li>4. User shares its image on professional or social service selected.</li> </ol>

## 8.4 Customer Support Use Case Model

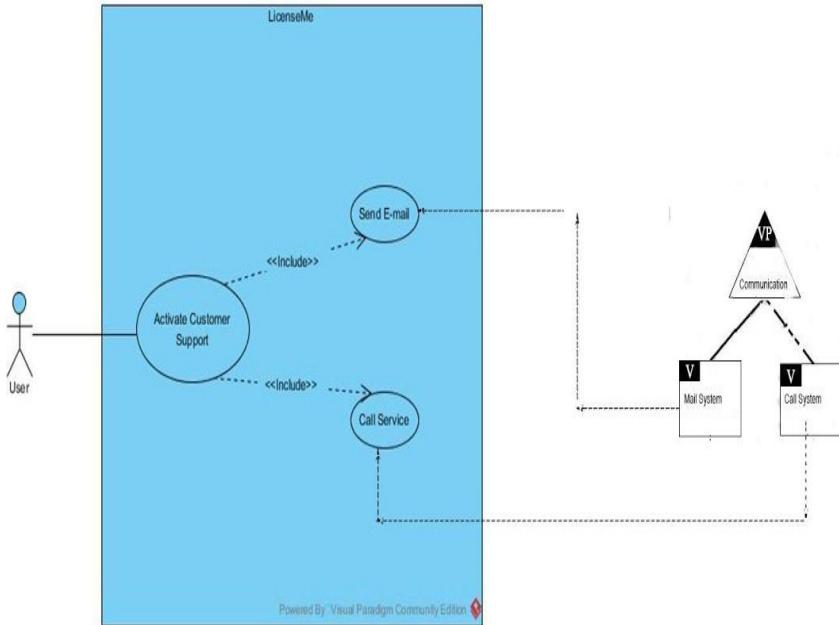


Figure 14: The figure showing the use case model for customer support

Users can contact with customer support via email and emergency call line. All users can benefit from daytime customer support. Emergency call line is only provided for professional usage.

Name	User uses communication channels.
Primary Actor	User
Preconditions	User should select call system or emergency hotline.
Main Scenario	<ol style="list-style-type: none"> <li>1. User enters the "LicenseMe" application and comes to select plan screen</li> <li>2. User selects Personal or Professional use packet or change plan to Professional use from Personal use.</li> <li>3. User clicks regular call system during working day hours.</li> <li>4. User communicates with authorities.</li> </ol>
Extensions	<p>User can change her/his plan type to Enterprise Use and use emergency hotline.</p> <ol style="list-style-type: none"> <li>2. User selects Enterprise use packet or change plan to Enterprise use from other use.</li> <li>3. User clicks the emergency hotline or regular call button at any time.             <ol style="list-style-type: none"> <li>3a. User clicks emergency hotline.</li> <li>3b. User clicks regular call line.</li> </ol> </li> <li>4. User communicates with authorities.             <ol style="list-style-type: none"> <li>4a. User communicates with authorities in any day time interval.</li> <li>4b. User communicates with authorities during working day hours.</li> </ol> </li> </ol>

## 8.5 Upload Image Use Case Model

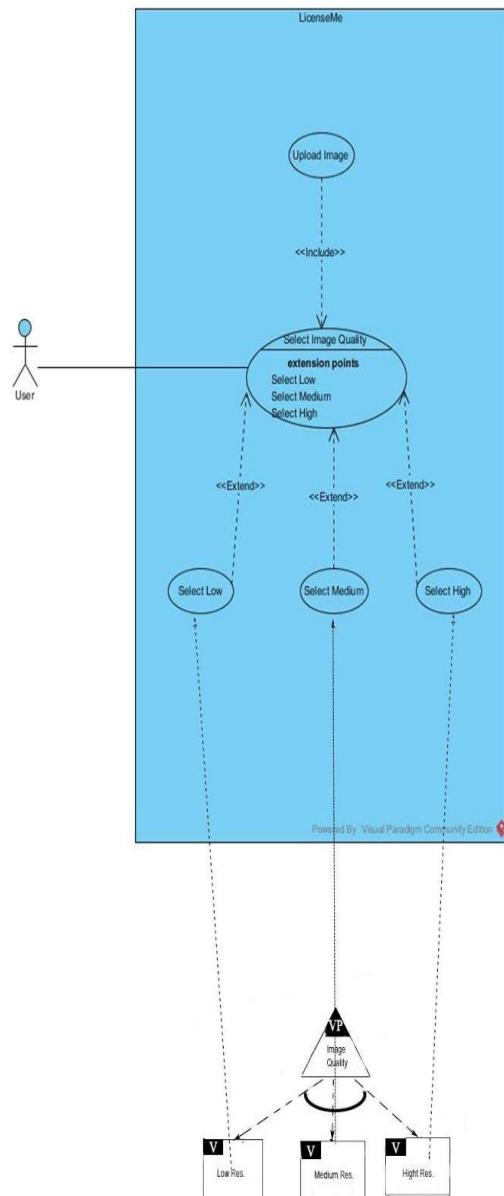


Figure 15: The figure showing the use case model for licensing

User can upload their photos to system for licensing them. Image quality is determined based on selected tier: low, medium or high for Personal, Professional and Enterprise tiers respectively.

Name	User uploads image for licensing process.
Primary Actor	User
Preconditions	User should select photo to upload.
Main Scenario	<ol style="list-style-type: none"> <li>1. User enters the “LicenseMe” application and comes to select plan screen</li> <li>2. User selects plan.             <ol style="list-style-type: none"> <li>2a. User selects Enter select Personal use.</li> <li>2b. User selects Enter select Professional use.</li> <li>2c. User selects Enter select Enterprise use.</li> </ol> </li> <li>3. User clicks camera button to select image to upload.             <ol style="list-style-type: none"> <li>3a. Low image quality for Personal use.</li> <li>3b. Medium image quality for Professional use.</li> <li>3c. High image quality for Enterprise use.</li> </ol> </li> </ol>
Extensions	-

## 8.6 File Management Use Case Model

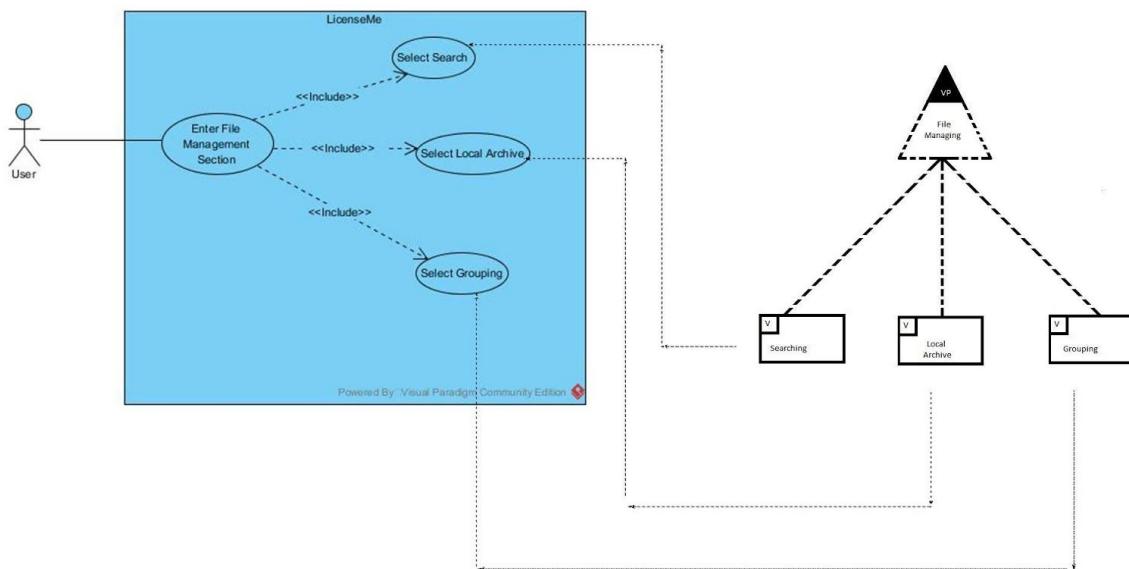


Figure 16: The figure showing the file management option

Users can manage their photos. They have 3 different file management options which are searching photos, saving in local archive, and grouping the photos. Each option depends on account type.

Name	User uses file management systems
Primary Actor	User
Preconditions	User's packet should be Enterprise
Main Scenario	<ol style="list-style-type: none"> <li>1. User enters the “LicenseMe” application and clicks camera button.</li> <li>2. User search for a specific picture</li> <li>3. User groups his favorite pictures</li> <li>4. User saves his photos to local archive</li> </ol>
Extension	-

## 8.7 Pricing Options Use Case Model

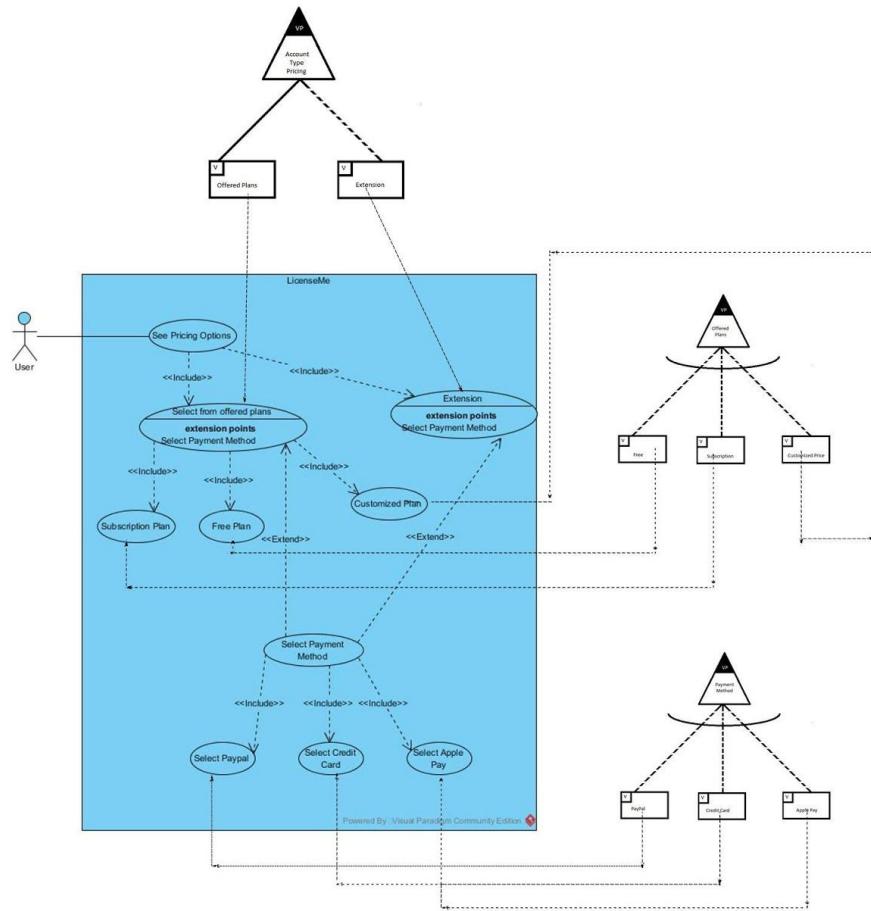


Figure 17: The figure showing the pricing for licensing photos

This model includes three different variation point which are connected to one use case model. This model describes seeing pricing options such as offered plans, extension, and payment methods. User can see the pricing options which are offered plans and extension through the LicenseMe. These pricing options are created after the selection of account license type. There different options are provided in offered plans which are subscription plan, free plan, and customized plan. Users can make their payments by selecting the payment method which includes PayPal or Credit Card. Users in professional or enterprise tiers can pay via Apple Pay.

Name	User uses file management systems
Primary Actor	User
Preconditions	-
Main Scenario	<ol style="list-style-type: none"> <li>1. User enters the “LicenseMe” application and clicks “Change option”.</li> <li>2. User selects from listed packages             <ol style="list-style-type: none"> <li>2a. User selects “Personal” package</li> <li>2b. User selects “Professional” package</li> <li>2c. User selects “Enterprise” package</li> </ol> </li> <li>3. User sees the offered plan             <ol style="list-style-type: none"> <li>3a. User uses free plan</li> <li>3b. User uses subscription plan</li> <li>3c. User uses customized plan</li> </ol> </li> </ol>
Extension	<p>User can change her/his package to “Professional” or “Enterprise” package and use payment methods.</p> <ol style="list-style-type: none"> <li>1. User makes payment from listed payment methods.             <ol style="list-style-type: none"> <li>1a. User selects PayPal</li> <li>1b. User selects Credit Card</li> <li>1c. User selects Apple Pay</li> </ol> </li> </ol>

## 9. Prototype

Following is the walkthrough of LicenseMe application.

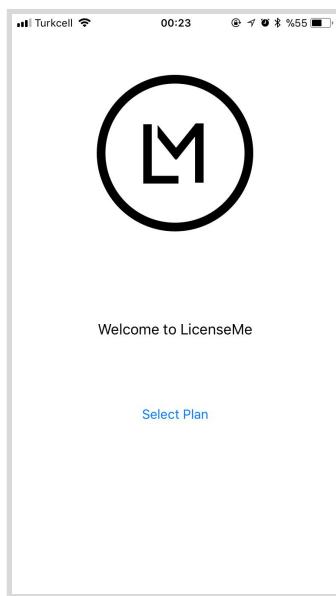


Figure 18: The figure showing the main screen of LicenseMe

Users can select their tier in homepage, by pressing ‘Select Plan’ button.

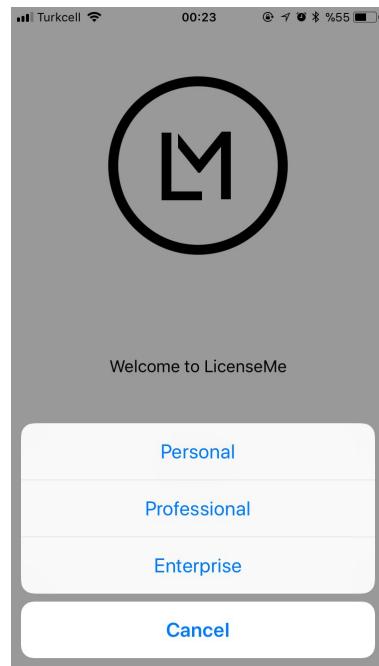


Figure 19: The figure showing the selection of user type

Three plan options are presented in an action sheet. Following is the view when Personal plan is selected.



Figure 20: The figure showing the personal tier screen

Since features in Personal plan are limited, some functionalities are disabled. Users can upload new photos to license using the camera button in the bottom toolbar. When users press the camera button, an image picker view is presented.

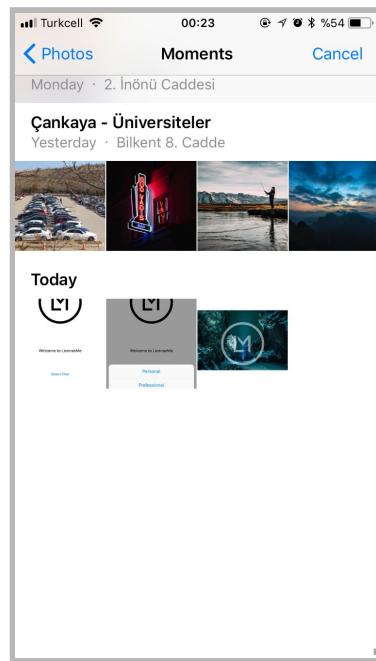


Figure 21: The figure showing the selection of uploaded photo

When users finish uploading a new image, the view is updated and new image is watermarked automatically.



Figure 22: The figure showing the licensing stage of uploaded photo

Users can share their licensed images using the share button located in left-bottom share button located in bottom toolbar. When users press share button, an action sheet with different share options is presented.

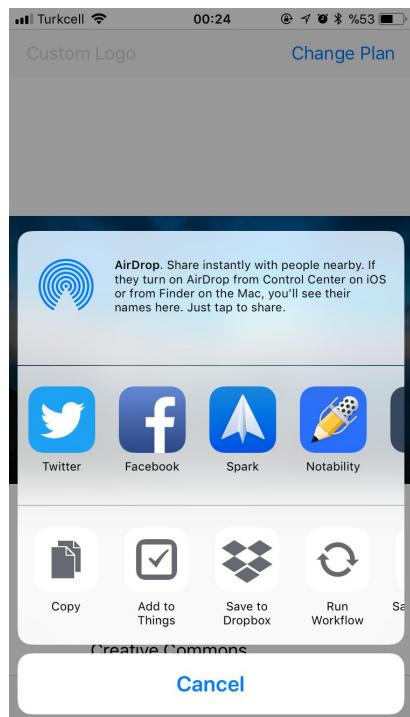


Figure 23: The figure showing the sharing for licensed photo

Users can call customer support using the button with a handset icon. When users press this button, a pop-up view will be presented.

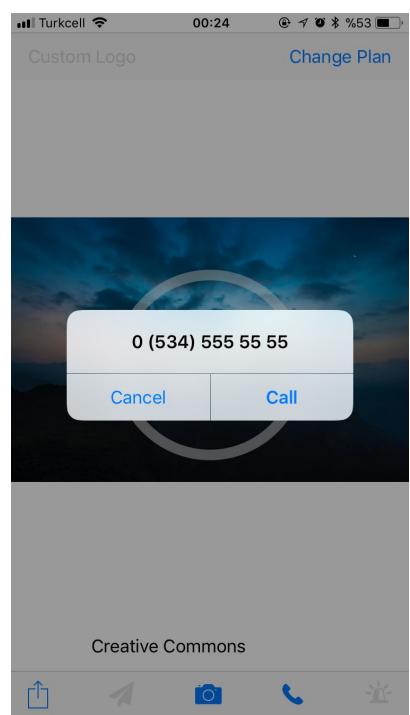


Figure 24: The figure showing the daytime customer support of application

Users can change their tier by pressing Change Plan button located in top-right corner. When users press this button, this view will be dismissed and the view with Select Plan button will be presented

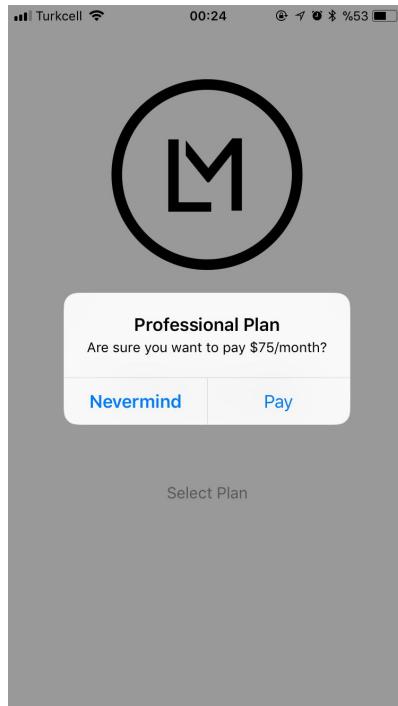


Figure 25: The figure showing the payment confirmation

If the user select Professional Plan, a pop-up confirming the payment will be presented and if the user selects to pay, Professional Plan will be activated.

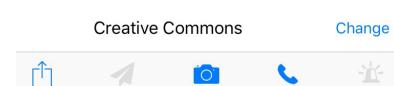
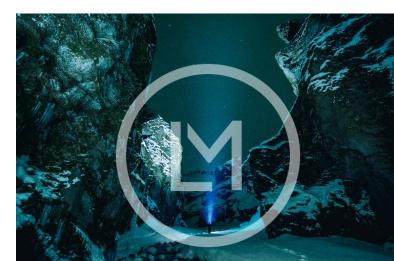


Figure 26: The figure showing the professional tier screen with additional features

In professional plan, additional features all enabled, such as changing license and choosing a custom logo for watermarking. When users press Change button to change the license, an action sheet with available license choices will be presented.

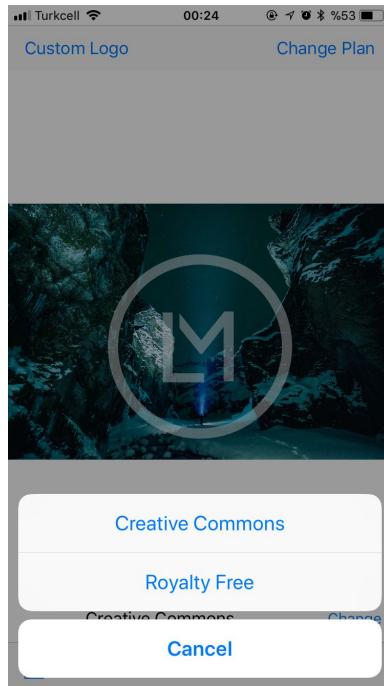


Figure 27: The figure showing the licensing type after uploading the photo

Users can also select their custom logo for watermarking. When users press custom logo button, an action sheet with available logos will be presented. When users pick their logo, the view is automatically updated and future images are watermarked using the new logo.

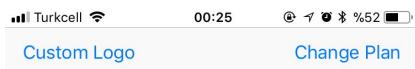


Figure 28: The figure showing the licensing with tier's own logo

When users pick Enterprise plan from Select Plan options, a pop-up confirming the upgrade will be presented.

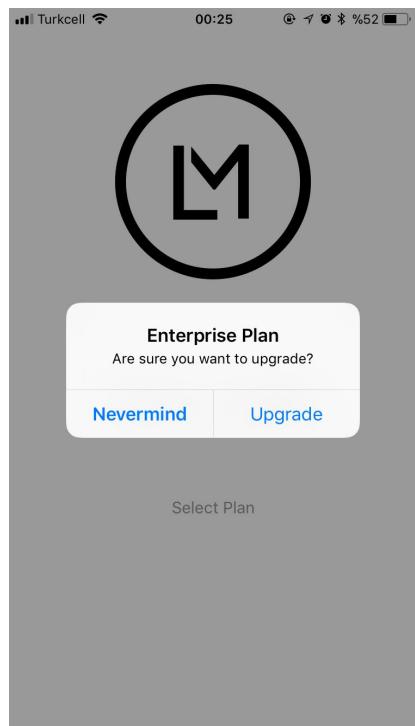


Figure 29: The figure showing the upgrading to enterprise tier

In Enterprise plan, all features are enabled.

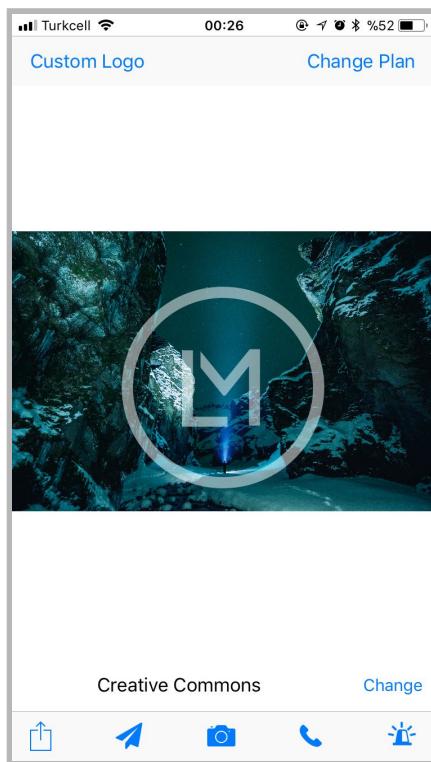


Figure 30: The figure showing the enterprise tier screen

When users press Change button to change license, every available license is presented in an action sheet.

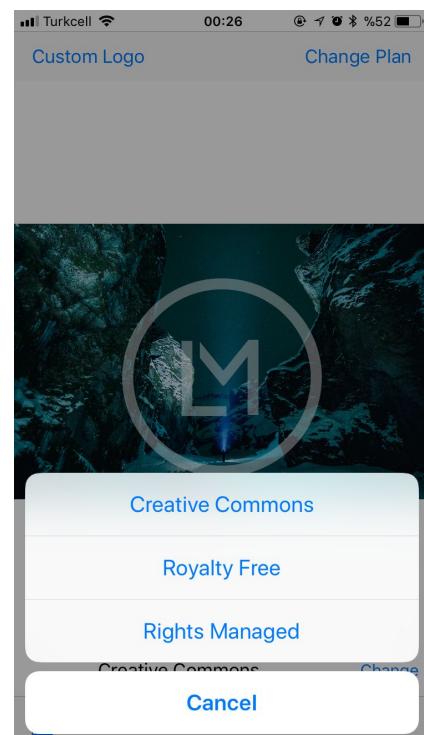


Figure 31: The figure showing the license type for enterprise

When users press emergency hotline button, an emergency hotline call pop-up will be presented. When users press professional share button with paperplane icon, an action sheet with sharing options will be presented. When users downgrade to Personal plan, a pop-up confirming the opt-out will be presented.

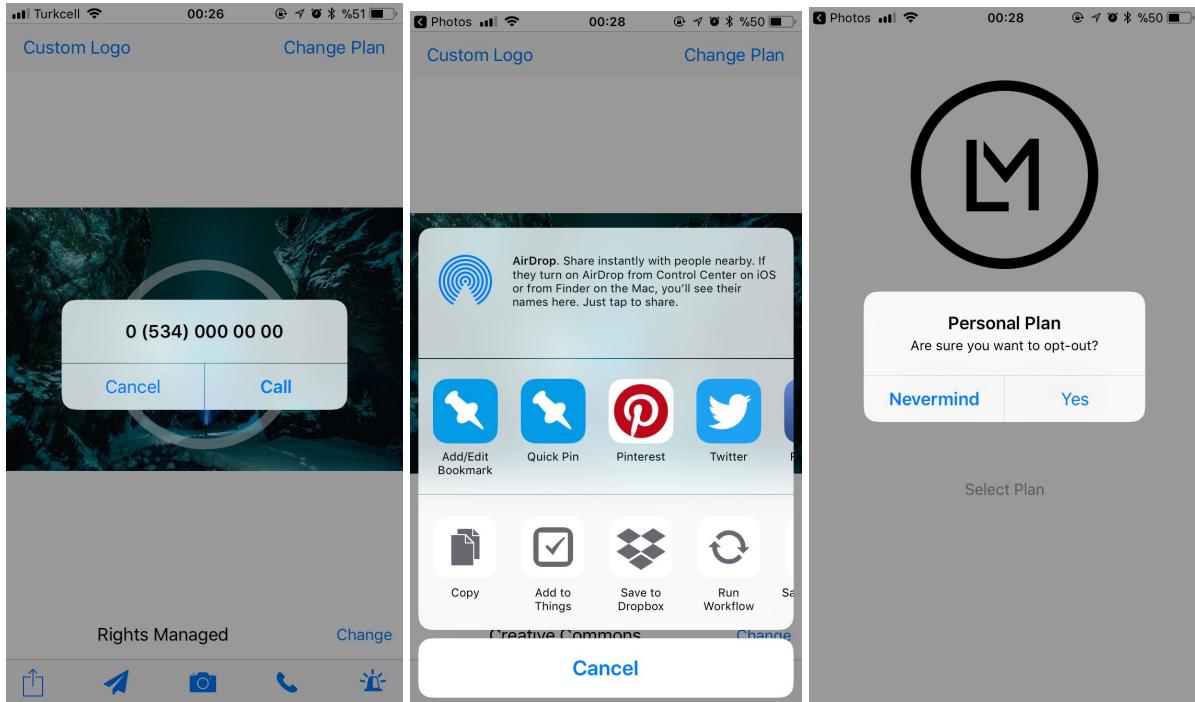


Figure 32: The figure showing the remaining screens of application

## 9.1 Software Reuse in Prototype

Software Reuse is implemented in code by storing a singleton value that represents selected tier, on device. This is achieved using `NSUserDefaults` and the singleton value is retrieved by `UserDefault.standard.string(forKey: "plan")` function call. When `UploadViewController` view is presented, a String variable indication the plan is also set. Using this variable necessary checks are carried out and several versions for different tiers are created from a single implementation.

An example usage of filtering based on plan is described below.

```
@IBAction func changeLicense(_ sender: Any) {
    let alertController = UIAlertController(title: nil, message: nil,
    preferredStyle: .actionSheet)

    alertController.addAction(UIAlertAction(title: "Creative Commons", style:
    .default, handler: { (action) in
        self.titleLabel.text = "Creative Commons"
    }))

    if plan == "Professional" || plan == "Enterprise" {
        alertController.addAction(UIAlertAction(title: "Royalty Free", style:
        .default, handler: { (action) in
            self.titleLabel.text = "Royalty Free"
        }))
    }
}
```

```

        if plan == "Enterprise" {
            alertController.addAction(UIAlertAction(title: "Rights Managed", style: .default, handler: { (action) in
                self.titleLabel.text = "Rights Managed"
            }))
        }

        alertController.addAction(UIAlertAction(title: "Cancel", style: .cancel, handler: nil))

        self.present(alertController, animated: true, completion: nil)
    }
}

```

In the code above, we add actions to alertController only if they are available in selected plan. Similar methodology is used throughout `UploadViewController` and `LandingViewController` classes.

We've also integrated custom tests that run automatically for a specific plan. When user changes plan to Enterprise, custom test functions start to run automatically. We've integrated these custom test systems to ensure that each variant works properly. To run the project, open the file with .xcodeproj extension on Xcode, then select a simulator and run.

## 10. Discussion

There are certain challenges we have met while improving the project. We handled these challenges by searching solutions from internet and books and evaluating what we have learned in the class. The challenges and what we have learned with this project are given below for each topic:

### 10.1 Requirements

We prepared a general set of requirements. This initial set included basic functionalities that enables us to create a minimum viable product. This is to make sure that we have a functioning application. We expanded this set to include any additional features that may increase customer satisfaction. This step was important since we wanted to make sure that we evaluate a comprehensive set of features when applying Kano Methodology.

### 10.2 Feature Model

As we have discussed in the class, feature model plays an important role for representing common, variable and product specific features in software product line. This feature diagram is used for demonstrating whole product line development process. In the project, after we discussed requirements for LicenseMe variants, we decided capabilities of each variant and drew the feature model based on our decisions. Since Kano methodology was applied after drawing feature model, we needed to change some of requirements after Kano methodology. However, first, Kano methodology should have applied then the feature model should be drawn. Because of the necessities of the project, we were not able to order them in this way which leaded some corrections in feature model. One of our assignments was based on drawing feature model but the requirements were given. This project was different from what we have learned from assignment. It was challenging to decide commonalities, variabilities and product specific features and draw feature model based on these requirements. It helped us to learn how to represent different type of features.

### 10.3 Kano Methodology

As we have learned in this course, Kano Methodology is applied to select the product features which yield high customer satisfaction. This methodology demonstrates that there are different types of customer needs which can be mapped to different requirement categories in Kano classification. Then, Kano methodology can be converted into SPLE requirements classification. In the project, we discussed on the possible requirements of the application for project description. Then, we prepared a questionnaire for Kano Methodology based on these requirements and it was answered by 5 group members. The problem was: since the questionnaire included some of already chosen requirements and there were no customers answering the questionnaire, Kano Methodology did not serve its main goal of selecting the product features yielding customer satisfaction. In addition, we learned a lot while preparing questionnaire. We have mentioned in class that there might be some problems related to Kano Methodology such as misinterpretations of questions, being tedious (each question was asked twice) etc. The questionnaire included 206 questions so we wrote questions as understandable and clear as possible for respondents. Selecting the right people as respondents was also crucial but it was not possible in our case so we tried to answer these questions from a customer point of view. This project helped us to experience and learn each stage of methodology. For requirements which were chosen as undesired, some of group members considered them as important requirements so they were added as questions. However, the results of questionnaire emphasized that the majority of the group members found them undesired. For example, for popup advertisements, some of group members said that it was required so we added it as a question but the Kano classification of it demonstrated that majority of group members found it undesired. Similar case was valid for internal variables. Some of group members considered them as important requirements for customers so they were added as questions but the results showed that they were internal variables. For example, for using google drive for professional photograph sharing, its question was added to questionnaire but results showed that majority of group members found it internal variable so its presence or absence did not affect customer satisfaction. Hence, although some of possible requirements were already chosen beforehand, Kano Methodology helped us to determine commonality and variability and eliminate some requirements.

### 10.4 Orthogonal Model

In this project, we used orthogonal variability model to document variability explicitly. Orthogonal model basically defines the variability of a software product line. There are many ways to demonstrate variability in orthogonal model. Use cases and variability model diagrams are used to demonstrate variation points and variants with functionalities. We answered the questions that what varies, why does it vary, how does it vary and for whom to provide adequate information. We learned that variability in the product line architecture may affect the architecture of system behaviour from software to hardware. Dependencies between functionalities and variabilities can be traced easily via orthogonal model. Dependencies also helped us to determine which tier includes which components. Also, if there is any change on LicenseMe application, we will know that which components will affect also from dependencies. In other words, we provide more communicable high – level abstraction and

traceability between artefacts. As we learned in the course, these types of links are necessary for application requirement engineering and keep information consistent.

While determining variation points and variant, we worked according to feature model of our system. Grouping the related systems into single use case was a difficult process for us because any mistake on the notation may also impede the other variants through constraints. Sometimes, some function seemed related too much. Thus, we spent a lot of time to separate them.

On another side, we could not find a proper tool to draw our orthogonal model directly. It can be achieved by some tools with indirect methods and unnecessary effort. We can say that SPLE Tool support aspect was not feasible for us. We drew use cases via Visual Paradigm and drew variability diagrams with Adobe Photoshop. This process cost us considerable amount of redundant time.

## 10.5 Prototype

LicenseMe application is built as a single product that changes its form automatically based on the selected plan. We've reused common UI elements from Interface Builder for every plan, instead of creating separate views for each plan. If a particular feature shouldn't be available in a plan, it's disabled by checking a singleton value that represents selected plan. We also integrated custom tests that run uniquely for each plan. We implemented this to make sure our features function properly.

Building a single adaptable application saved us development time. However, since LicenseMe is a mobile application, we weren't able to integrate feature model to the application. An IDE integration tool similar to IBM Rhapsody would enable us to change our feature model and make the app change automatically. As a result of our research, we've discovered that Xcode IDE doesn't have such integration. We've learned that, other applications with similar user base levels, can be implemented as a single product that adapts to a singleton value and this approach would save development time and cost.

# 11. Group Member Contribution

## Muhammed Çavuşoğlu

He mainly worked on the development of the prototype and test integrations. He also reported his parts in respective sections of project reports.

## Duygu Durmuş

She contributed mainly to kano methodology, feature model and variability configuration for each variant. In addition, she prepared discussion parts for these topics.

## Arif Can Terzioğlu & Özgür Can Erdoğan

They mainly developed use case scenarios and orthogonal model of high level architecture for each use case. They also worked on discussion parts of their topics.

## Mithat Orhan

He contributed to writing requirements and discussion parts of the report.