Project Overview

MY-NEIGH

ESCI *upf.*
International Business

Tools

Concepts

Feelings

Concepts

Sprint 1

- Project description

- Requirements:
  - Functional
  - Non-functional

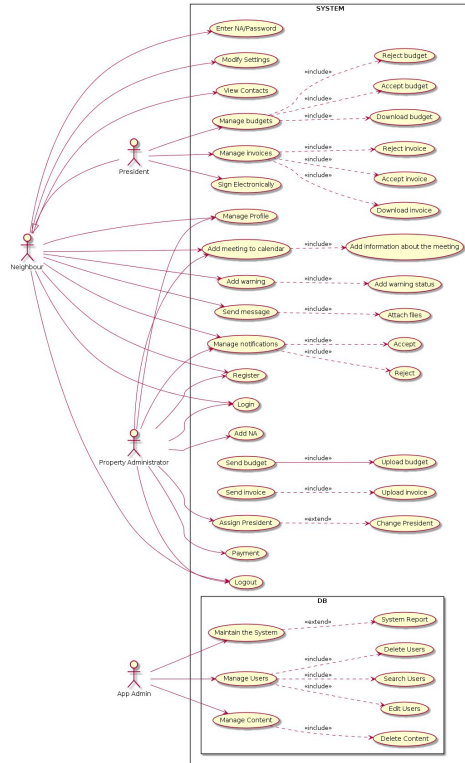| ID	REQ-F-2-02
Title	Property Administrator Information
Description	The property administrator information should have: • Name • Surname • Company Name • Email • Phone Number
Priority	H
Verification	
Parents	

ID	REQ-F-2-03
Title	PA Login
Description	The system shall manage a login system for the PA.The user can access with: • Email or username • Password In the future, the user will be logged automatically.
Priority	H
Verification	
Parents	

ID	REQ-F-2-04
Title	Verification Email
Description	The system will send a verification email to the PA email, after registration.
Priority	M
Verification	

4   Use Case : Remove event from calendar

The actors involved in this use case are: neighbour, president and the property administrator. They need to be able to remove a meeting or an activity from the calendar.

4.1   Events flow

4.1.1   Basic flow

The use case starts when the user has access to the system and is logged.

1. User must access to the calendar.
2. User must select a day, a month and a year from the calendar.
3. User must select an Event that is already in the calendar.
4. User must click "Delete Event".
5. The system will remove the Event and the users will not be able to see it.
6. Use Case Ends.

4.1.2   Alternative flow

- Move Event
    1. User selects another day to introduce the event.
    2. User return to the main page.
- Delete Error.
    1. The system will display an error message "The Meeting cannot be deleted". This can happen when a neighbour tries to delete a meeting added by the property administrator.
    2. The meeting will remain in the calendar.
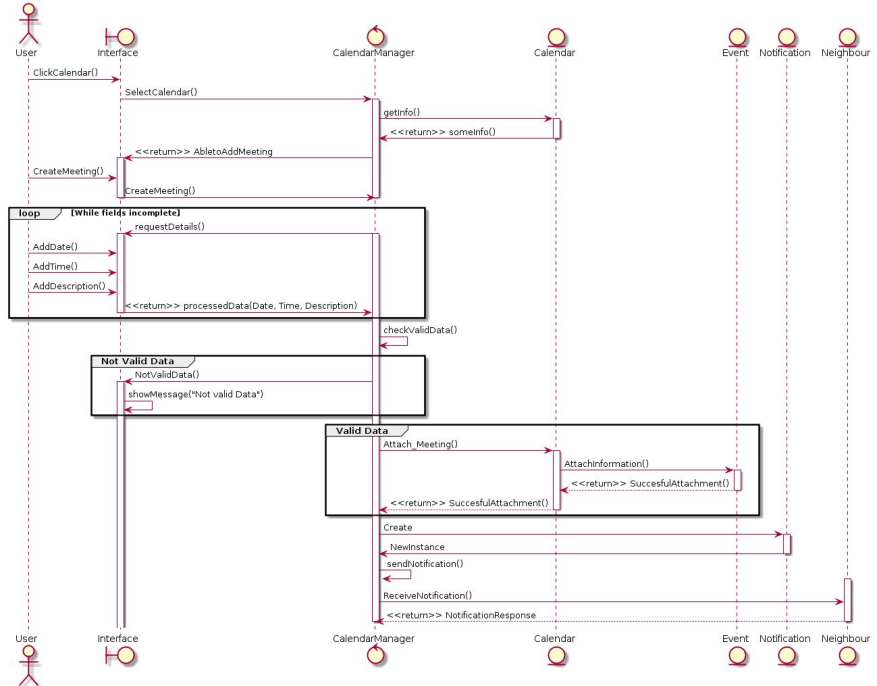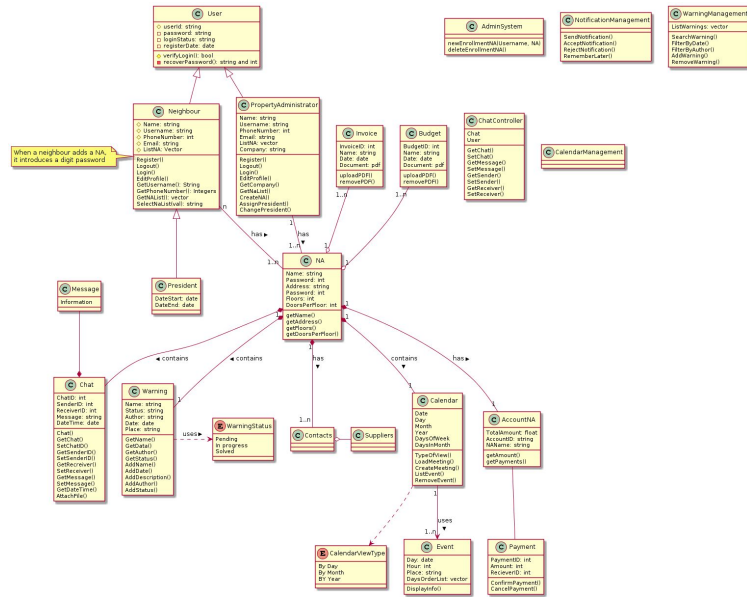
4.2   Pre-Conditions

- User Registration
- User Login
- Be in an neighbourhood association

4.3   Post-Conditions

- The meeting is removed from the calendar.

Concepts

Sprint 3

Class Diagram (left)

User
- userId : string
- password : string
- loginStatus : string
- registerDate : date
- verifyLogin() : bool
- recoverPassword() : string and int

AdminSystem
- newEnrollment(NAUsername: NA)
- deleteEnrollment(NA)

NotificationManagement
- SendNotification()
- AcceptNotification()
- RejectNotification()
- RememberLater()

WarningManagement
- ListWarnings: vector
- SearchWarning()
- FilterByDate()
- FilterByAuthor()
- AddWarning()
- RemoveWarning()

Neighbour
- Name: string
- Username: string
- PhoneNumber: int
- Email: string
- ListNA: Vector
- Register()
- Logout()
- Login()
- EditProfile()
- GetUsername(): String
- GetPhoneNumber(): Integers
- GetNAList(): vector
- SelectNAListVal(): string

When a neighbour adds a NA, it introduces a digit password

PropertyAdministrator
- Name: string
- Username: string
- PhoneNumber: int
- Email: string
- ListNA: vector
- Company: string
- Register()
- Logout()
- Login()
- EditProfile()
- GetCompany()
- GetNAList()
- CreateNA()
- AssignPresident()
- ChangePresident()

Invoice
- InvoiceID: int
- Name: string
- Date: date
- Document: pdf
- uploadPDF()
- removePDF()

Budget
- BudgetID: int
- Name: string
- Date: date
- Document: pdf
- upload(PDF)
- remove(PDF)

ChatController
- Chat
- User
- GetChat()
- SetChat()
- GetMessage()
- SetSender()
- SetSender()
- GetReceiver()
- SetReceiver()

CalendarManagement

Message
- Information

President
- DateStart: date
- DateEnd: date

NA
- Name: string
- Password: int
- Address: string
- Password: int
- Floors: int
- DoorsPerFloor: int
- getName()
- getAddress()
- getFloors()
- getDoorsPerFloor()

Chat
- ChatID: int
- SenderID: int
- ReceiverID: int
- Message: string
- DateTime: date
- Chat()
- GetChat()
- SetChatID()
- SetSenderID()
- SetReceiver()
- GetMessage()
- SetMessage()
- GetDateTime()
- AttachFile()

Warning
- Name: string
- Status: string
- Author: string
- Date: date
- Place: string
- GetName()
- GetDate()
- GetAuthor()
- GetStatus()
- AddName()
- AddDate()
- AddDescription()
- AddAuthor()
- AddStatus()

WarningStatus
- Pending
- In progress
- Solved

Contacts

Suppliers

Calendar
- Date
- Day
- Month
- Year
- DaysOfWeek
- DaysInMonth
- TypeOfView()
- LoadMeeting()
- CreateMeeting()
- ListEvent()
- RemoveEvent()

AccountNA
- TotalAmount: float
- AccountID: string
- NAName: string
- getAmount()
- getPayments()

CalendarViewType
- By Day
- By Month
- BY Year

Event
- Day: date
- Hour: int
- Place: string
- DaysOrderList: vector
- DisplayInfo()

Payment
- PaymentID: int
- Amount: int
- ReceiverID: int
- ConfirmPayment()
- CancelPayment()

Sequence Diagram (right)

Participants: User, Interface, CalendarManager, Calendar, Event, Notification, Neighbour

- ClickCalendar()
- SelectCalendar()
- getInfo()
- <<return>> someInfo()
- <<return>> AbletoAddMeeting
- CreateMeeting()
- CreateMeeting()

loop [While fields incomplete]
- requestDetails()
- AddDate()
- AddTime()
- AddDescription()
- <<return>> processedData(Date, Time, Description)

- checkValidData()

Not Valid Data
- NotValidData()
- showMessage("Not valid Data")

Valid Data
- Attach_Meeting()
- AttachInformation()
- <<return>> SuccesfulAttachment()
- <<return>> SuccesfulAttachment()

- Create
- NewInstance
- sendNotification()
- ReceiveNotification()
- <<return>> NotificationResponse

Concepts

Sprint 4

MY-NEIGH

Test Procedures

Ref : **BBDI19-GR-04**
Version: v1.0
Date: June 11, 2019
Pag. 4 of 9

ESCI upf.
School of International Studies

2   TP-01 : Add meeting

In this Test Procedure we will verify all possible scenarios related to the addition of a meeting.

Step	01
Description	System Start
Input	The system must be available and user shall be logged with a username and password
Output	System booted properly, without errors
Comments	

Step	02
Description	Main Page Access
Input	The system shows the main page interface of a Neighbourhood association
Output	Main Page without error
Comments	

Step	03
Description	Calendar Access
Input	When the user clicks the calendar should be open
Output	The system opens the calendar window.
Comments	

Step	04
Description	Default view
Input	The calendar window is displayed
Output	The default current month view must be displayed
Comments	

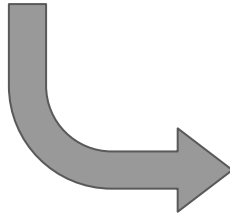| Step	05
Description	Adding new event
Input	User clicks to new meeting
Output	The system displays a new window with empty spaces that the user shall fill: • Name • Place • Day/Month/Year • Description
Comments	

Bitbucket

📁 /

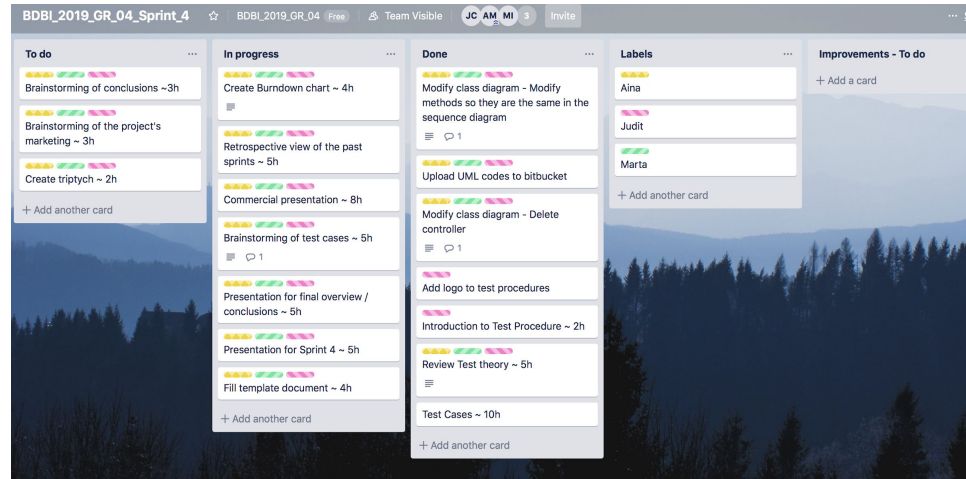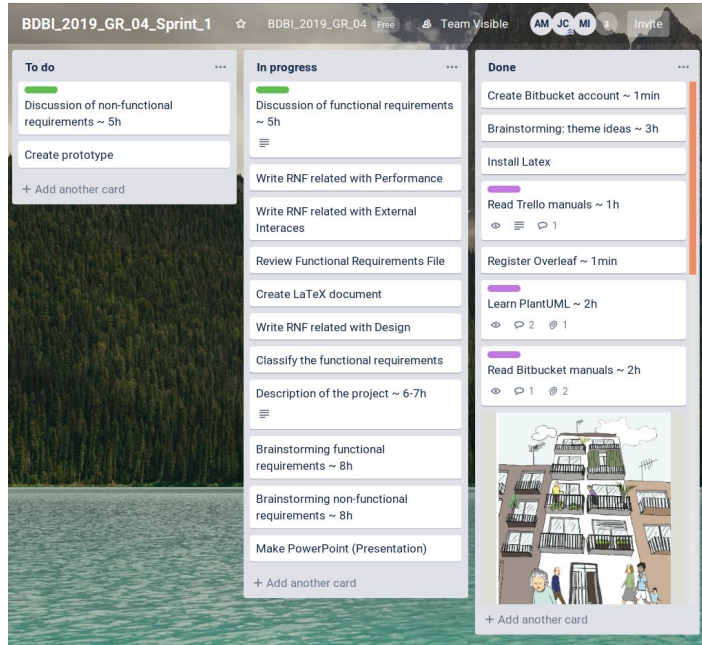| Name	Size	Last commit	Message
📄 BDBI-Practicum-Template-Presentation.tex	2.19 KB	1 minute ago	BDBI-Practicum-Template-Presentation.tex
📄 BDBI-Practicum-Template-Requirements	3.13 KB	5 days ago	BDBI-Practicum-Template-Requirements created online
📄 Description of the project (draft)	4.03 KB	yesterday	Description of the project (draft)
📄 README1.md	621 B	2019-05-01	This is the explanation file to describe our initial idea and
📄 Requirements (drafts)	819 B	2019-05-03	Requirements (drafts)
📄 reqs_functionals.tex	9.85 KB	yesterday	reqs_functionals.tex
📄 reqs_nfunctionals.tex	5.33 KB	3 hours ago	reqs_nfunctionals.tex

📁 /

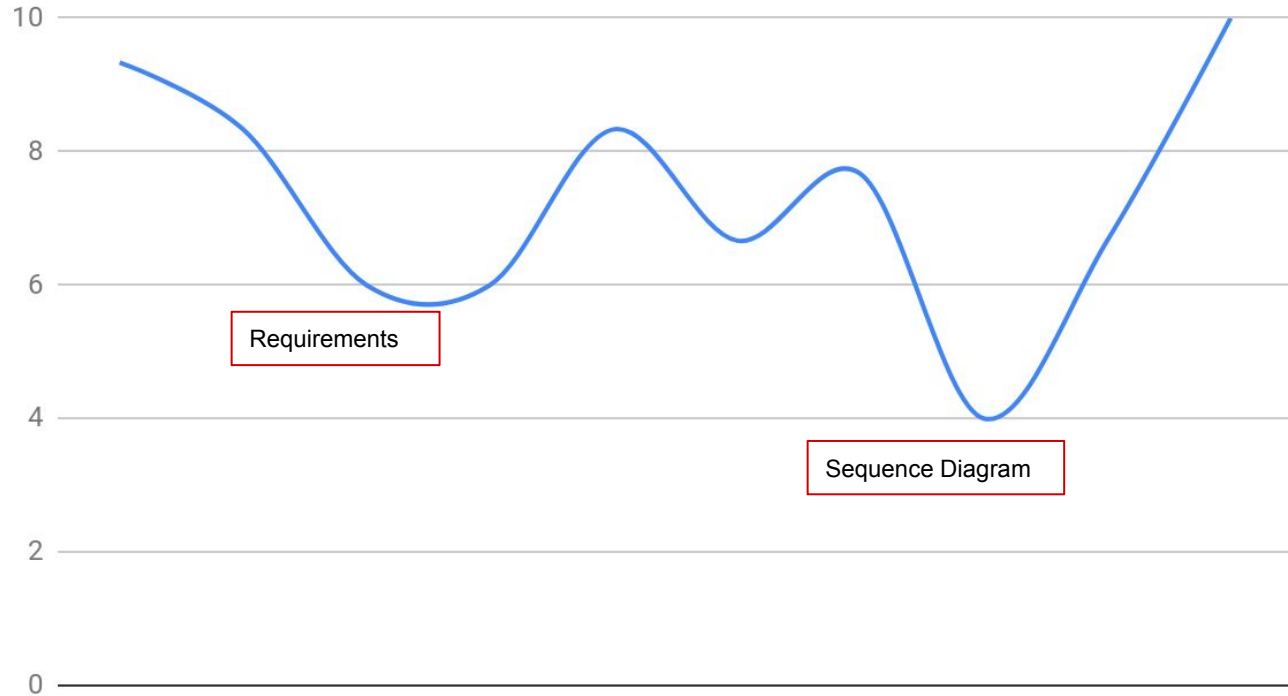| Name	Size	Last commit	Message
📁 Documents		41 minutes ago	Added Test case for Notification Management
📁 Presentations		7 hours ago	restructured Presentations
📁 UML		7 hours ago	renamed folder
📄 README1.md	621 B	2019-05-01	This is the explanation file to describe our initial idea and in which ste

Tools

Trello

Feelings

Feelings of MY-NEIGH

This is a survey to find out how the team members felt during the project. Different topics are presented and a mark from 1 to 10 should be given to them.

Being:
1 - you have hate doing the topic
10 - you have loved the topic

Project Idea and Initial Setup *

	1	2	3	4	5	6	7	8	9	10
	○	○	○	○	○	○	○	○	○	○

Project Description *

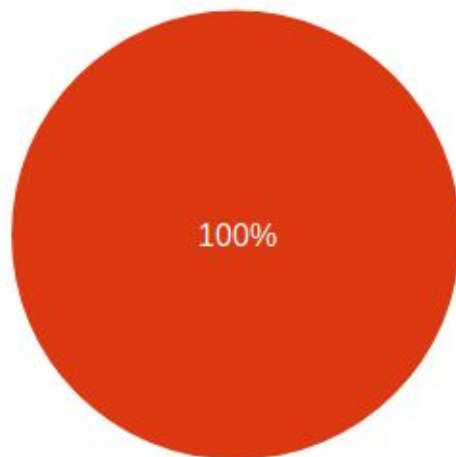|	1	2	3	4	5	6	7	8	9	10

Feelings

Project Feelings

Your favorite sprint?

3 respostes

- Sprint 1 - Requirements
- Sprint 2 - UCSD
- Sprint 3 - Design
- Sprint 4 - Test Procedures

100%

Main Conclusions

1. Work with repositories: git commit, git status

2. We have learned an agile methodology: SCRUM!

3. We have learned a lot of new concepts!

4. We have really enjoyed the project!!

Main Conclusions

Thank you for your attention!