Amaury Morales
Josue Torres

Toj Tu Yeetel

Repository: https://github.com/amcamcamc/Toj-Tu-Yeetel

Description: Toj Tu Yeetel (Healthy crop in Mayan language) is an information system for farmer people where they can track their crops, they can report on the pests and plagues that their crops are suffering, as well as view on a map and in a 'feed' different plague reports in greenhouses or nearby crops, so that they can take action before they can fall victim to a nearby plague and to prevent the economic loss that results from disposing of contaminated product.

Functional Requirements

#FR00.- The user can register in the system.

#FR01.- The user will be able to log in to the system.

#FR02.- The system should display in the feed reports sent by other users.

#FR03.- The system should display on a map reports sent by other users.

#FR04.- The user will be able to access the map and see the information of his crops.

#FR05.- The user can add a crop to track it.

#FR06.- The user will be able to observe all the information of his crop.

#FR07.- The user may mark a crop as "harvested".

#FR08.- The user may make multiple reports of multiple plagues.

#FR09.- The system must have four main sections (inicio, cosechas,reportes and mapa).

-#FR10.- The system must operate the database using Mongoose.

-#FR11.- The system must use the OpenStreetMap API to complete the #FR03.

-#FR12.- The map will select the user's desired location.

Amaury Morales
Josue Torres

+#FR13.- The system should show the locations of the user's registered crops.

Non-Functional Requirements

#NFR00.- The experience of users regarding the use of computer systems should be a "sufficient" level.
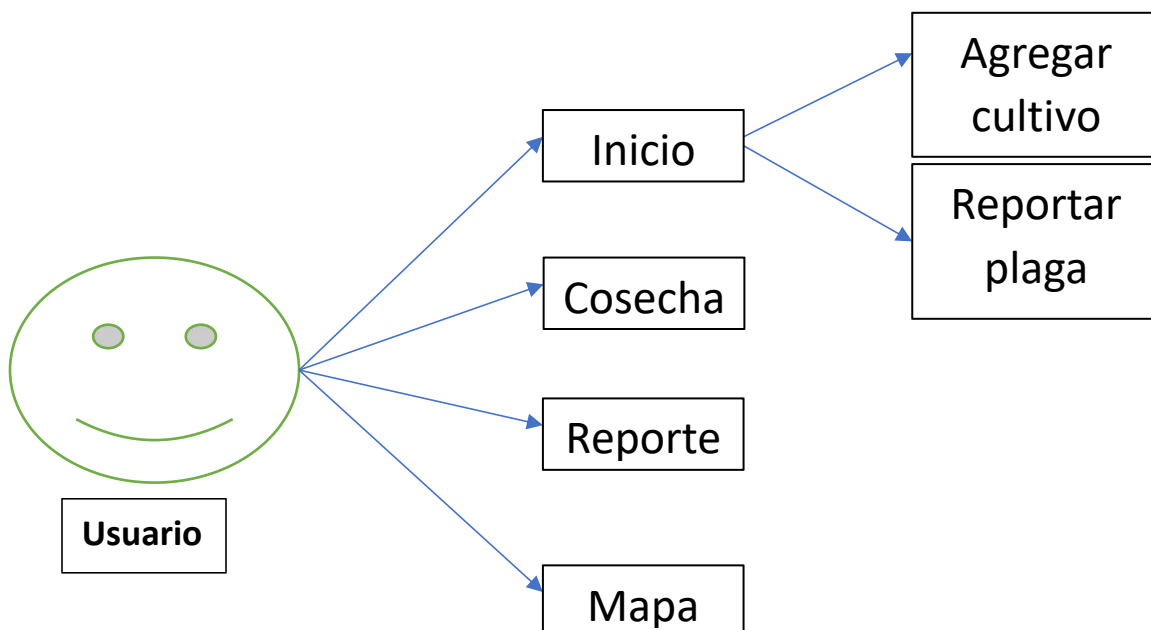
#NFR02.- The system shall handle/create exceptions of the following types:

• Input values by the user.

• Database.

#NFR03.- The system should be considered to move to a mobile platform in the future.

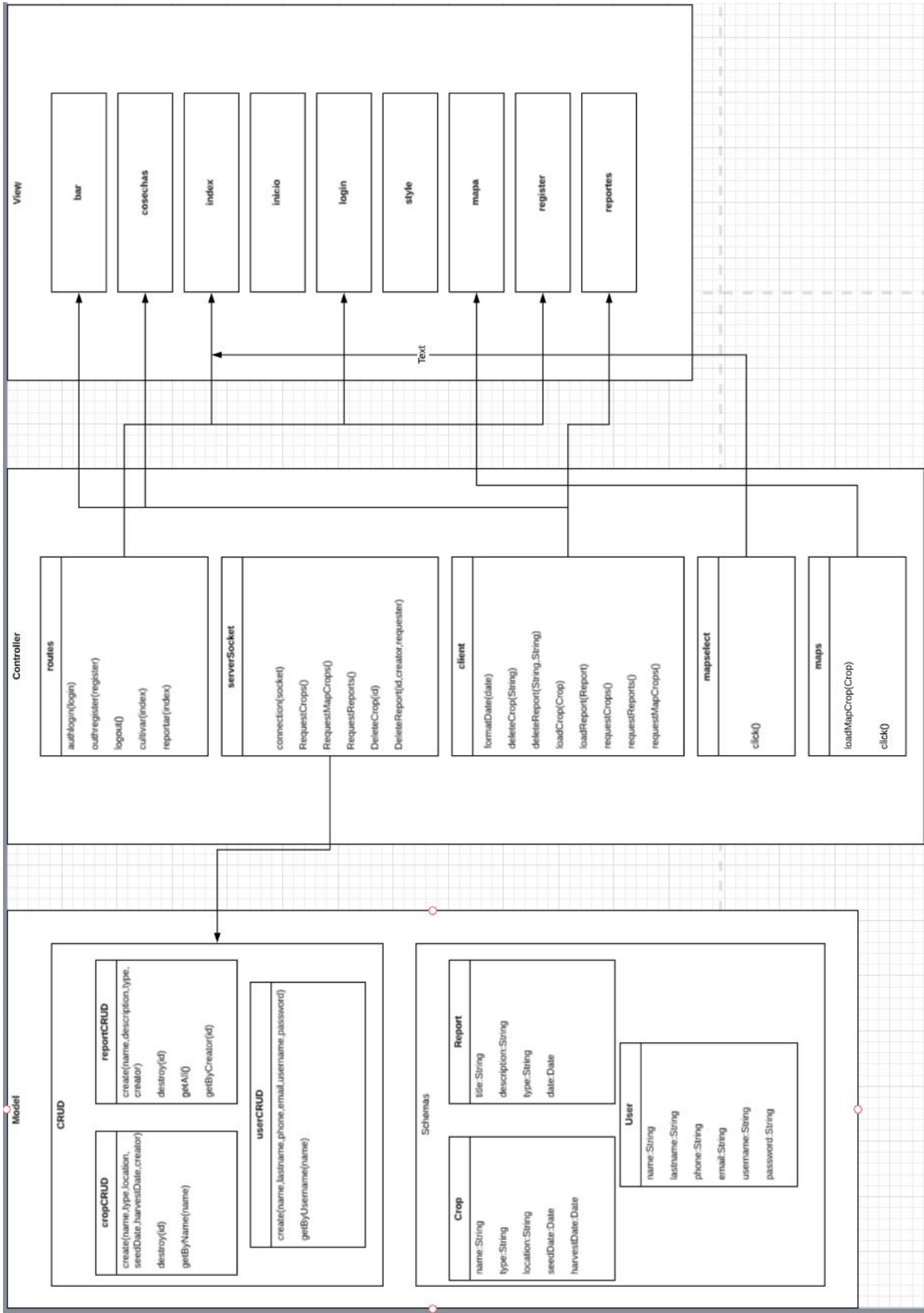#NFR04.- A standard point will be used as a center on the map.

use case diagram

Amaury Morales
Josue Torres

Use cases

Name	CU-01 Display reports
Actor(s)	Farmer(user)
Description	The user will be able to view in the feed reports from other users
Preconditions	*The user must be logged in. *The user must be in the "reportes" tab.
Normal flow	1. The user scrolls to view reports from other users.
Exceptions	

Name	CU-02 Create a report
Actor(s)	Farmer(user)
Description	The user will be able to make reports on found plagues.
Preconditions	*The user must be logged in. *The user must be in the "inicio" tab.
Normal flow	1. The user enters a title to the report. 2. The user selects a plague type from the report. 3. The user writes a plague description. 4. The user presses the "Reportar" button.
Exceptions	1. If the user does not have an internet connection when pressing the "Send Report" button the information will not be displayed and a browser message will be displayed. 2. If the user does not complete all the required fields when presses the "Submit Report" button, the error will be displayed, and the user will be able to fill them.

Name	CU_03 Register crops
Actor(s)	Farmer(user)
Description	The user will be able to register crops to track them.
Preconditions	The user must be logged in. The user must be in the tab "inicio".
Normal flow	1. The user enters a name for the crop. 2. The user selects a crop type.

	3. The user defines the planting date of the crop. 4. The user defines the harvest date of the crop. 5. The user enters the location of the crop. 6. The user presses the "registrar" button
Exceptions	1. If the user does not have an internet connection when pressing the "registrar" button, the information will not be displayed and a browser message will be displayed. 2. If the user does not complete all the required fields when presses the "registrar" button, the error will be displayed, and the user will be able to fill them.

Name	CU_04 Crop tracking
Actor(s)	Farmer(user)
Description	The user can track and view information about each one of his crops.
Preconditions	The user must be logged in. The user must have registered at least one crop. The user must be in the "Mapa" tab.
Normal flow	1. The user click on a "pin" image on the map. 2. The user can observe the name,type, planting and harvest dates of the crop.
Exceptions	

Amaury Morales
Josue Torres

View

bar

cosechas

index

inicio

login

style

mapa

register

reportes

Text

Controller

routes
auth.login(login)
auth.register(register)
logout()
cultivar(index)
reportar(index)

serverSocket
connection(socket)
RequestCrops()
RequestMapCrops()
RequestReports()
DeleteCrop(id)
DeleteReport(id,creator,requester)

client
formatDate(date)
deleteCrop(String)
deleteReport(String,String)
loadCrop(Crop)
loadReport(Report)
requestCrops()
requestReports()
requestMapCrops()

mapselect
click()

maps
loadMapCrop(Crop)
click()

Model

CRUD

reportCRUD
create(name,description,type,
creator)
destroy(id)
getAll()
getByCreator(id)

cropCRUD
create(name,type,location,
seedDate,harvestDate,creator)
destroy(id)
getByName(name)

userCRUD
create(name,lastname,phone,email,username,password)
getByUsername(name)

Schemas

Report
title:String
description:String
type:String
date:Date

Crop
name:String
type:String
location:String
seedDate:Date
harvestDate:Date

User
name:String
lastname:String
phone:String
email:String
username:String
password:String

Amaury Morales
Josue Torres

Task	Responsible	value
Create the repository	Amaury	1
Create the main UI, segmented by tabs	Amaury	3
Create the UI to add a new crop.	Amaury	1
Create a base form Pane.	Amaury	1
Create the UI to report a plague.	Amaury	1
Create the UI to track crops.	Amaury	1
Handle the transitions in the different UI's.	Amaury	3
Valid the input information from the 'add new crop UI'.	Amaury	4
Valid the input information from the 'report a plague UI'.	Amaury	4
Create Exceptions	Amaury	5
Pass the reports to the view	Amaury	2
Show the registered crops in the UI	Amaury	4
Pass the registered crops to the view	Amaury	4
Show the crop information on the UI	Amaury	4
Show the reports in the UI.	Amaury	5
Set up the Mongo Database	Amaury	11
Configure the project for Glitch, and other utilities	Amaury	13
Create the Entities' classes	Amaury	3
Create CRUD methods for the Entities	Amaury	6
Documentation	Josue	5
Testing	Amaury/Josue	4/4
Explanatory code comments	Josue	1
Tools research	Amaury	10

Total:**100** points

Contribution:

Amaury: 90/100=90**%**

Josue: 10/100= 10**%**