**קבוצה .21**

חברי הקבוצה :

 פיראס אלגואברה .319146684

 מוחמד אבו מדיעים .315744821

 עלאא אלדין חונדוקלו .204161491

 אסמאעיל אלנסאסרה .209525922

שם המתרגל : אלכסנדר גפטר.

הפרויקט: אתר אפיון ויזום חכם.

**Github link:** <https://github.com/aladiha/nehool-project>

user stories member:

state

create a project done

send request to join a project done

view requests that I send done

view rejected requests done

view accepted requests done

accept a request to join a project done

view requests that has been sent to me done

request to leave a project done

view my projects done

project maneger:

accept membership request done

reject membership request done

view all membership requests done send request for a member to join my project done

חלוקת חברי הקבוצה :

 הסקרם מאסטר: פראס אלגואברה.

 הלקוח : אלכסנדר גפטר..

 :DPA עלאא אלדין חנדוקלו.

 מפתיח : מוחמד אבו מדיעים.

 מעצב: אסמאעיל נסאסרה .

מפגש ראשון:

נפגשנו בספרייה למשך 4 שעות מ 12:00 עד ל 16:00 , תוכן המפגש היה על חלוקת עבודה כלומר לקבוע מי הסקרם מאסטר , אחראי בסיס נתונים , הלקוח, המעצב

והמפתח. בנוסף סיכמנו ספרנט ראשון סגרנו פינות שהיו בספרינט הראשון .

וקבענו מטרה למפגש הבא שנבוא מוכנים בשאילות ונעזור אחד לשני.

וגם חלוקת תרשימי עיצוב לפי הדרישות.

מפגש שני:

נפגשנו בספרייה למשך 4 שעות מ 10:00 עד ל ,14:00 בהתחלת המפגש סגרנו פינות מי

שהצטרך עזרה בחלק שלו עזרנו לו ואז קבענו מטרות חדשות למפגש הבא.

ההתחלנו ללמוד איך משתמשים ב Jenkins איך לחבר אותו לפרויקט באמצעות pipeline

וב Jira איך לבצע work log וכו...

לכל STORY USER גזרנו כמה SUBTASKS שאותן רוצים לממש , לדוגמה עצוב הדרישה ,

מימוש הדרישה עצמה, בדיקות וכדומה.

מפגש שלישי:

היה המפגש האחרון לפני סיום הספרינט , דנו בפגישה על מה הבעיות שהיו לנו במשך

התכנות או עצוב ,דנו מה היה עדיף לעשות לפני או מה יהיה טוב לאחרי דרישה מסוימת, נתנו כמה סיכומים עד ליום כלומר מה צריך לשפר לספרינט הבא , ומה כידי לעשות או

לשייך (בקצרה לספרינט השני) .

קשיים:

 בהתחלה היה לנו קשה לממש ה requests מבלי לסבך עניינים בסוף עלינו על זה

בהגדרת בסיסי נתונים מתאמים שעזרו לנו הרבה בשימוש.

 בגלל שלכל אחד יש מערכת שעות שונה ,היה קשה קצת לסדר לפגישות שבעיות.

 היה לנו חוסר בתקשורת אחד לשני שזה גרם לאיבוד מעכב בינינו לכן קבענו שעה

ביום לתיאום ציפיות.

 במהלך העבודה הייתה בעיה בחיבור בסיס הנתונים שקרס מסיבה כלשהיא אבל

היה לנו בסיס נתונים לגיבוי שחברנו אותו במקומו.

 עבודה עם Jira in work log בהתחלה לא הבנו מה צריך לעשות בדיוק ואיך

לקבוע זמן מסוים לכל דרישה אבל בסוף הבנו איך בעזרת המתרגל.

project free אבל כשדברנו עם

Jinkins בהתחלה עשינו את זה דרך

 חיבור ל

המתרגל התברר שהיינו צרכים לחבר באמצעות pipeline וזה מה שעשינו.

כעקרון אלו היו הבעיות הכי עיקריות שנתקלנו בהן.

ניהול סיכונים:

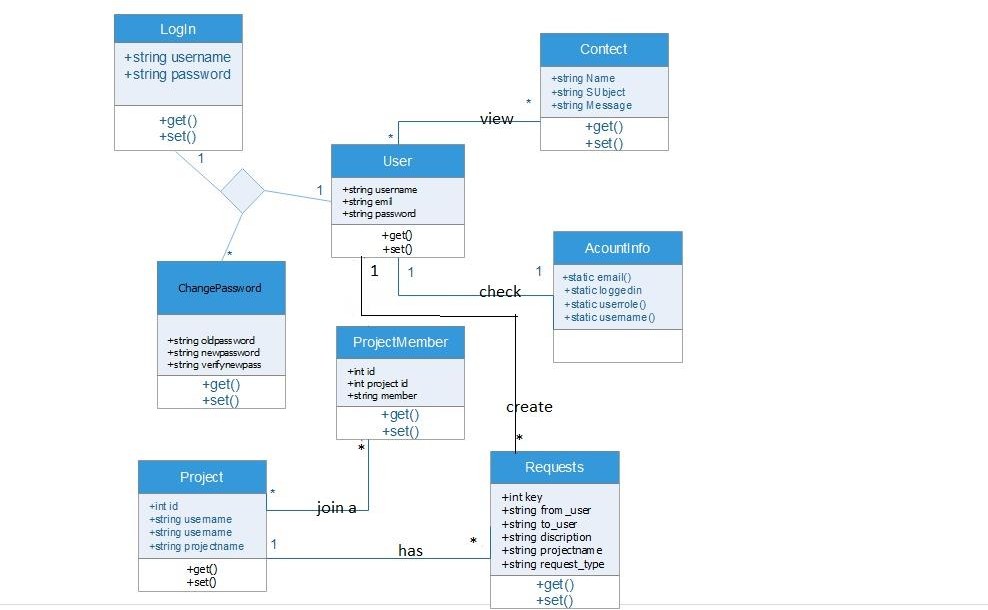
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **5**  **Catastrophic** | **4**  **Major** | **3**  **Moderate** | **2**  **Minor** | **1**  **Negligible** |
| **5**  **Almost certain** | **-** | **-** | - | Visitors More than expected | - |
| **4**  **Likely** | Connection to database fail | Authorize as an member and get admin  access | - | Accept a member in one project and get him added in  all projects | - |
| **3**  **Possible** | Authorize as an admin and get member access | System failure | Leave a certain project and get deleted from all projects | - | Create a project and get multi creations |
| **2**  **Unlikely** | Create same project name | **-** | **-** | Not see requests that I send | Not see requests That I get. |
| **1**  **Rare** | Try to log in and get an error | Not show a type of a request | A user added more than once to the same project | Try to log out and not succeed | **-** |

**Key:**

|  |  |
| --- | --- |
| **Extreme** | Extreme risks that are likely to arise and have potentially serious consequences requiring urgent attention |
| **Major** | Major risks that are likely to arise and have potentially serious consequences requiring urgent attention or investigation |
| **Medium** | Medium risks that are likely to arise or have serious consequences requiring attention |
| **Minor** | Minor risks and low consequences that may be managed by routine procedures |

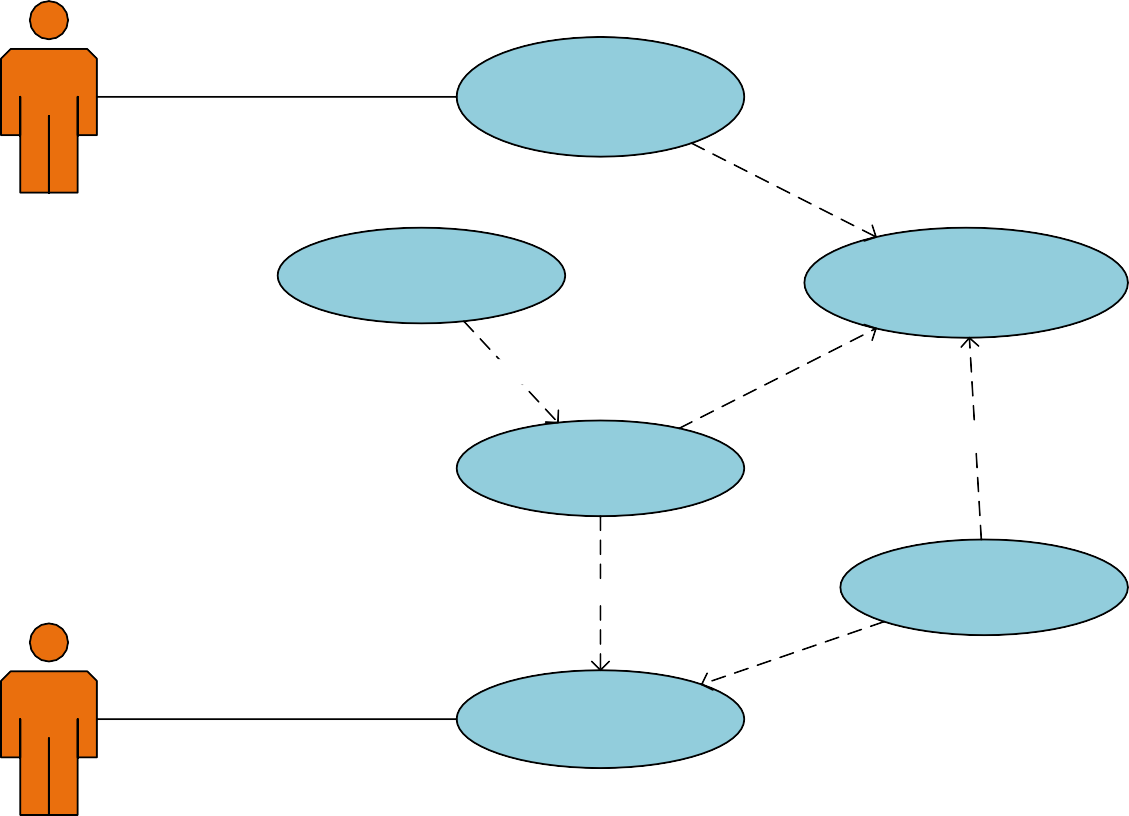
# UML

## Class diagram



Use Case

### Create project and Requests (accept reject management)

create project

project maneger

leave project

<<include>>

send request to join my project

<<extend>>

<<extend>>

accept request

<<extend>>

<<extend>>

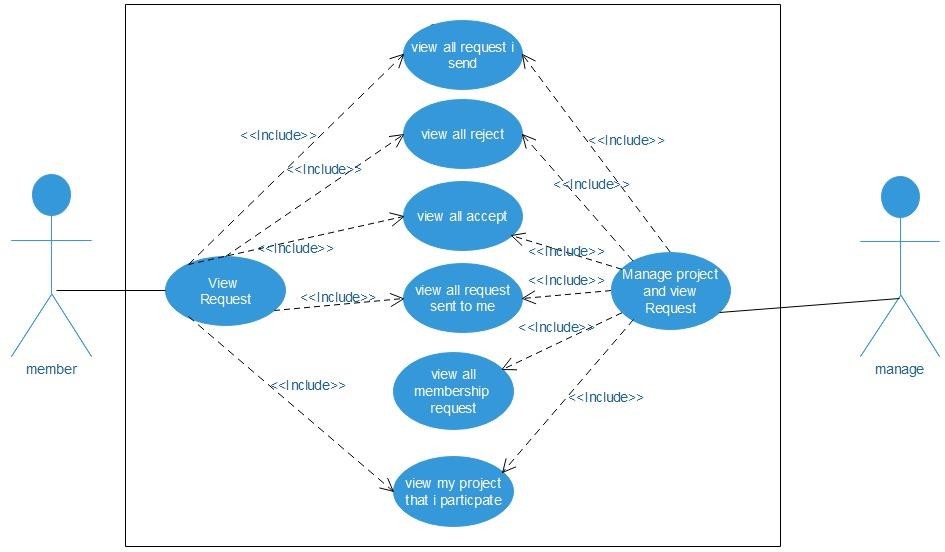
reject request

send request to join a project

<<extend>>

project member

View Requests



# Activity Diagram

View my requests



Request

My Request

Show Requests

Table

### Create and send a request

Request

New Request

user name project name Description

Request Type

send to join

a project

send to leave a

project

send to join my

project

### Accept or reject a request

Request

view request

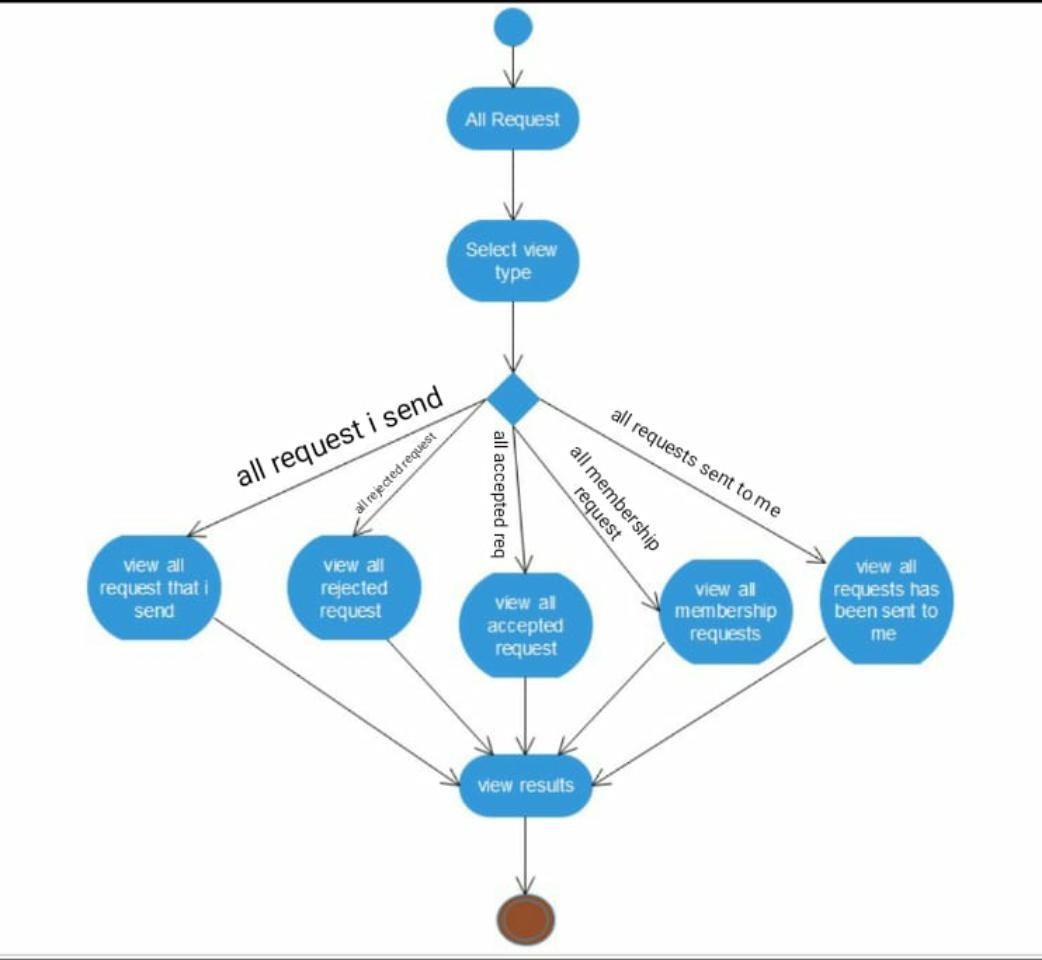
Show table recent

requests

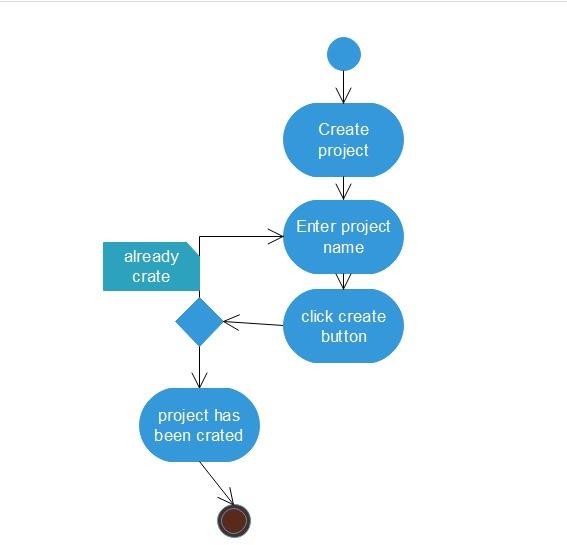
reject request

accept request

View Requests



Create Project



**Testing**

What are we testing:

1. Create a project

We create a project and check if it does exist in the database if it exist the function returns true if not false.

1. send a request to join a project

we create a request and send it with the receiver name we check the database return true if the request exists the function returns true if not false.

1. View requests (accepted and rejected) that I send

We use a display function that uses the database to get the logged in user requests and returns true if it gets available info and false if not.

1. Accept or reject a request

We create a new request and accept it or reject it with the receiver account if its states changed we return true if not we return false.

1. Leave request

We create a request and send it we remove the user from the database return true if the user is removed with the specific project name if yes the function returns true if not it returns false.

Notice: there are more requirements that has the same testing functions as the ones mentioned above.