

IT-314 Software Engineering Course Project Kick-off



Group : 27 College Programming Club

IDs	Name
202001276	OM CHALODIYA
202001268	MANSARA FENIL LALITBHAI
202001272	ARCHIT SINGH
202001265	SANGANI KISHAN PARESHKUMAR
202001267	BHATT MEET DHAVALBHAI
202001426	VADODIA HARSHIT JAGDISHBHAI
202001256	TUSHAR DHAR
202001241	RIKI ANILBHAI PARIKH
202001433	MODIYA DEEP ASHOKBHAI
202001445	PATEL KRUNAL MAHESHBHAI

Needs:

The need for a project solution for the programming club website is to provide an online platform for members to share and collaborate on their programming projects, as well as to promote the club and its activities to a wider audience.

A programming club website can help to build a community of like-minded individuals, provide resources and opportunities for members to learn and grow, and improve the club's visibility and impact.

1. Event Management:

- To manage the programming club related activities like event scheduling, event registration, attendance tracking, etc. effectively and efficiently.

2. Information sharing:

- To provide information about the club's goals, events, etc to the potential stakeholders (club members) and users of the system.

3. Efficient Interaction:

- To facilitate smooth interaction between the system and students.

4. Secure Access for club members:

- Members of the club should be able to access the system securely through login ID, allowing them to post about upcoming events and schedules of that particular event.

5. Encourage participation:

- To encourage the participation of students in various upcoming events.

6. Track performance:

- To track performance of the students in the past events.

7. Consistency:

- To keep the information stored by the system consistent.

8. Integrity:

- To maintain the integrity among the details stored by the system.

Key features:

1. User registration:

- The system supports the registration facility for its stakeholders and users.

2. User login:

- The system supports the login facility for its stakeholders and users.

3. Club events and activities:

- A section to promote upcoming events and activities related to the programming club, as well as to provide information about past events.

4. profiles:

- The system supports the profiles of the users.

5. User-friendly interface:

- A clean, easy-to-use interface that makes it simple for members to navigate the site and find the information they need.

6. Mobile responsiveness:

- The website is optimized for viewing on a variety of devices, including smartphones and tablets.

7. Security:

- The system ensures security and protection against threats, to ensure the safety of the user's personal information.

8. Feedback:

- Feedback section is also important for programming clubs to improve session content.

9. Authentication:

- The system supports the authentication of stakeholders and users.

10. Attendance tracking:

- The system supports the feature to track the attendance of students in various events like programming sessions and contests.

11. Real time:

- The system supports the real time updation of the information.

Functional requirements:**1. User registration:**

- The system supports the registration facility for its stakeholders (club leaders and members) and users.

2. User login:

- The system supports the login facility for its stakeholders (club leaders and members) and users.

3. Event registration:

- The system supports the event registration facility for its users.

4. Event de-registration:

- The system supports the event de-registration facility for its users.

5. Event scheduling:

- The system only allows its stakeholders (club leaders and members) to schedule the events.

6. Attendance tracking:

- The system tracks the attendance of the users, who have registered for the particular event.

7. View events:

- The system allows all the users to view the past as well as upcoming events list.

8. Search event:

- The system allows all the users to search for a specific event.

9. Volunteer hours tracking:

- The system keeps track of the volunteer hours of club members.

10. Budget and fund tracking:

- The system keeps track of the budget and fund related activities of the club.

11. View attendance records:

- The system only allows its stakeholders (club leaders and members) to view the attendance records.

Non functional requirements:

1. Performance:

- The system should be quick and responsive, with minimal lag times for a better user experience.
- The system should be able to handle high volumes of concurrent users and transactions.

2. Scalability:

- The system should be designed to accommodate growth and be easily scalable to meet increasing demands.
- The system should be good enough to handle a vast database of the users and the property details.

3. Availability:

- The system should have a high availability rate and be designed to minimize downtime.
- The system should have robust disaster recovery mechanisms in place.

4. Security:

- The system should have strong security measures to protect sensitive information, such as user passwords and payment information.
- The system should comply with relevant data protection regulations.

5. Usability:

- The system should be user-friendly and intuitive, with a clear and simple navigation structure.
- The system should be accessible and usable for users with disabilities.

6. Interoperability:

- The system should be compatible with a range of devices and browsers.
- The system should be able to integrate with other systems, such as payment gateways, APIs, etc.

7. Reliability:

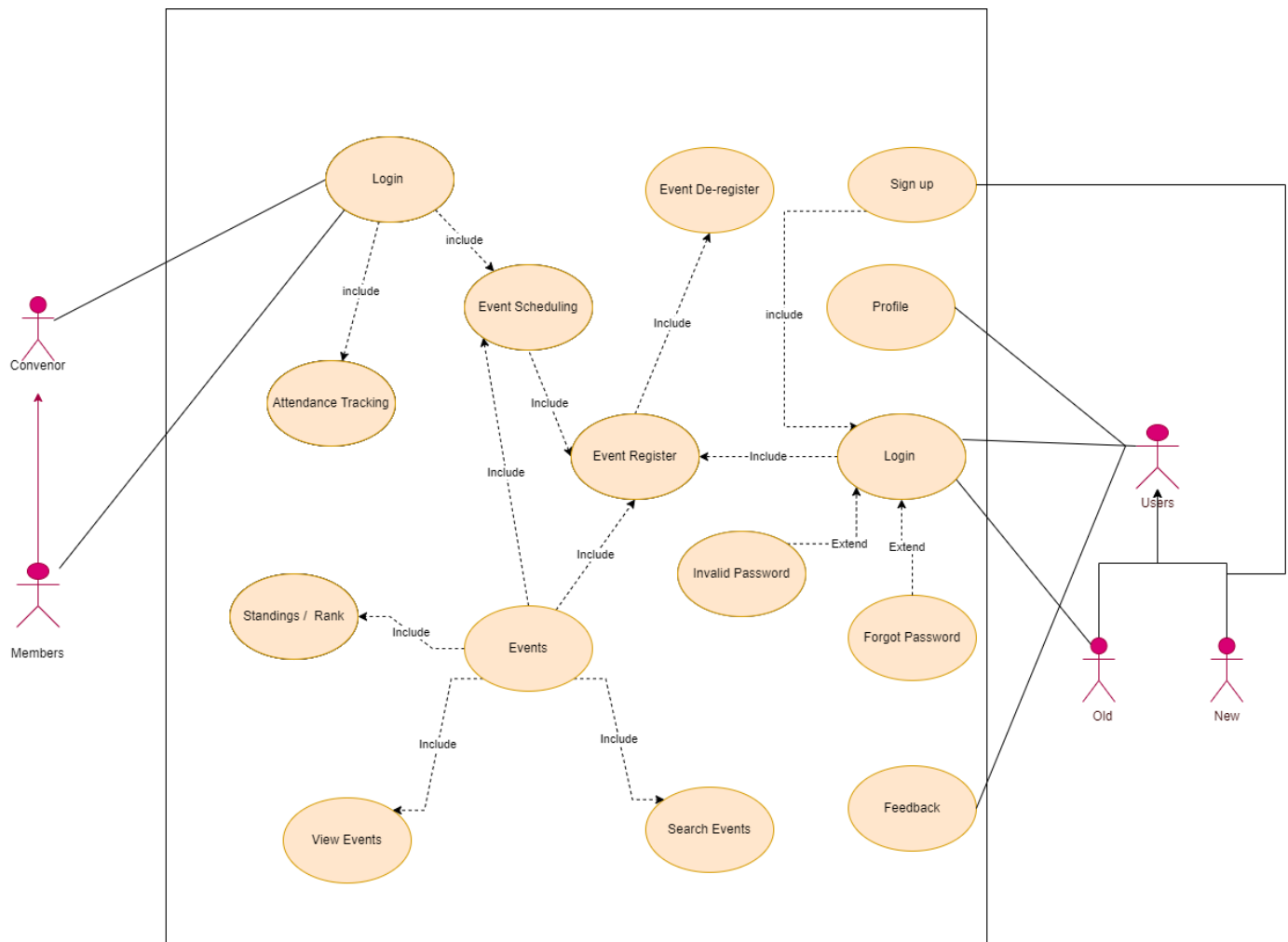
- The system should have a high level of reliability, with minimal errors and bugs.

8. Maintainability:

- The system should be designed to be easily maintainable, with clear documentation and an organized code structure.
- The system should have the capability to be updated and improved over time.

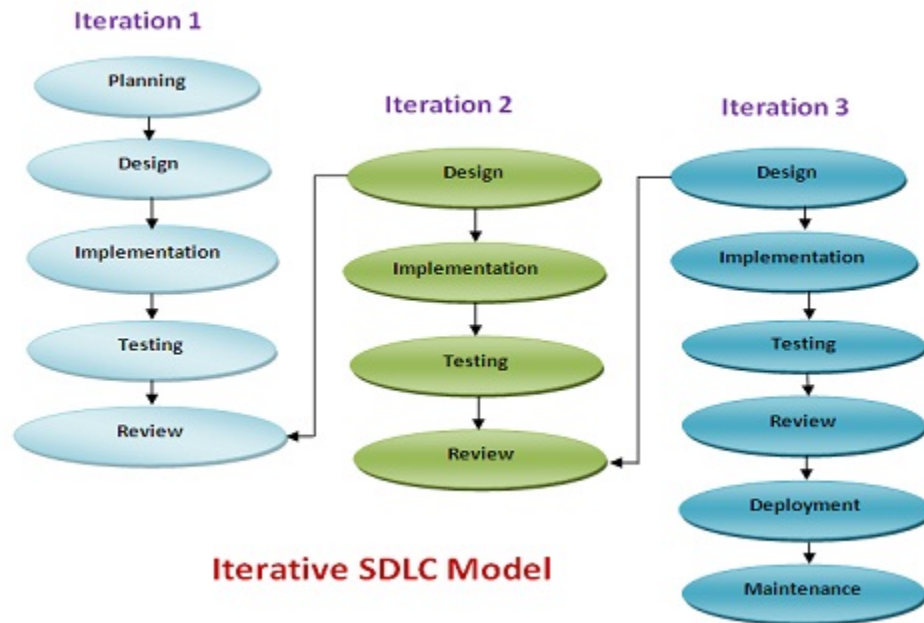
Portability, risk, authentication, robustness, privacy etc, are some of the other non-functional requirements.

Use Case Diagram:



Process model best suited for our course project:

After reviewing the needs and requirements of the project, we decided that the **iterative process model** is the best suited for our project. There are many reasons to use an iterative process model.



- The iterative model is easily acceptable to ever-changing needs of the project.
- It takes limited time spent on documentation and extra time on designing.
- Easier to test and debug during smaller iterations is easy.
- Risk analysis during each iteration.