Software Engineering Project Report

Project title: PES Times – A web portal for clubs and events in PES with a forum for book exchange

Team: SEB-3

Department, Semester, Section: Computer Science and engineering, 6th semester, B section

Members:

1PI13CS099 : Neha M Kalibhat

1PI13CS085 : Manisha Rachel Dawson

1PI13CS091 : Mohit Mayank

1PI13CS097 : Nagasundar

1PI13CS098 : Navneet Singh

1PI13CS100 : Niket Raj

1PI13CS104 : Parikshit Maheshwari

1PI13CS108 : Prafful Um

1PI13CS125 : Rohan Ds

1PI13CS126 : Romasha Suman

1PI13CS141 : Sharath Kp

1PI13CS161 : Smitha

1PI13CS199 : Anirudh Agarwal (Team Leader)

GitHub repository: https://github.com/anirudhagar13/PES-Portal.git

Index

Feasibility Study

Revision History of Feasibility Study

Revision #	Date	Author	Where	Change	Approved by
1	31-01-16	Team	Entire Report	Baselining	Team
2	07-02-16	Neha	Section 3	Added the current scenario of book exchange in 3.1 and modified 3.3 to suit the context	Team
3	07-02-16	Neha	Section 7	Extended time allotted for work integration	Team
4	07-02-16	Parikshit	Section 4.3	Have elaborated the book exchange feature.	Team
5	07-02-16	Parikshit	Section 4.4	Have corrected the database requirement	Team
6	07-02-16	Anirudh	Section 4	Added a vague timeline for project development along with a mention of product's self-sufficiency.	Team
7	07-02-16	Manisha	Section 6	Rectified the changes by mentioning the competitors, talked about student exclusivity and stating the book exchange feature only once.	Team
8	07-02-16	Anirudh	Section 8	Reframed old and added new issues.	Team

1. Problem Statement

PES Times. A unified PES University web portal for the students and by the students. From an information center for upcoming events to a social platform for the students to discuss and socialize, PES Times is an attempt at making all the scattered Facebook pages and groups for different clubs and events obsolete. This portal tries to solve the problem of students missing out on events and meetings just because they weren't properly notified or forgot to check the respective pages on Facebook. Additionally, it provides a means for student discussions, interactions and suggestions, regardless of the year or department the student belongs to. It takes a crack at making the signup process for different occasions easier and user-friendly. PES Times also makes an effort to organize the numerous clubs in PES University. Different clubs have different pages where they can upload photos, create events and keep a calendar of upcoming events, and a lot more. Finally, buying the books second hand from Campus Mart or asking around the seniors is always a real pain. This PES portal aims to ease this process with the "Book Exchange" feature, which is like a forum where users can put up books they are trying to sell and interested students can contact them. The object of PES Times is to create a globalized and unified portal so that the students do not need to look for anything more, anywhere else.

2. Executive Summary

PES University has a plethora of clubs, each having numerous events round the year. While these groups and their activities make the college diverse, it's obviously not easy to keep track of each one of them. This leads to interested students missing out because of no dedicated system to provide them with properly organized notifications. This is where PES Times comes into play. The primary objective of this web portal is to provide the students with every information about every clubs and events, whether past, ongoing or upcoming, at their fingertips.
Another key problem that students face is buying the books. Either they need to go around asking the seniors for help, or wait forever in the confined space of the Campus Mart. The
"Book Exchange" part of the portal allows students to share books among each other at their own convenience.
Summarily, PES Times is a platform, created by and dedicated to the students, which tries to make their lives way more convenient at the university.

3. Current Systems and Processes

Every club uses social platforms like Facebook, Google etc., which is maintained by its respective admin (Club leader). Students need to follow these pages and track websites, in order to be updated.

3.1 Current Operations

As mentioned, clubs have their own platforms like pages, groups in social networking sites and software applications to post their upcoming events and other activities. Clubs maintain their own system (like forms and manual registration), to enable registrations to the events. To know and participate in any events organized by clubs, a student needs to explicitly check corresponding platform.
Discussions about the events are done through social networking sites and queries and clarifications are solved in person or over phone. Promotional messages and posters are sent amongst students through WhatsApp or email.
When it comes to the management of books, the current scenario is, newly joined students buy their textbooks online and buy photocopies of notes in campus while seniors who are about to pass out, sell their books at shops or simply dispose them. Only a few students have the privilege of obtaining books from seniors they know or are in touch with.

3.2 Physical Environment

There are no specific hardware requirements on such applications as they are mainly web-based but a browser is a must. Generally, there is a web server which will host the application and provide services to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP). Pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to text content. Server will connect to and query the database using server-side scripting language.
In these applications there is usually an administrator who will view and manage the Website configuration through a simple Web interface. He can also create a database to store application services data, such as membership and roles information. Clients will access these services made available by a server through some software mainly web browsers.
There also exists a Notification Services application which will generate notifications, and distribute them to subscribed clients on regular interval. The application developer will determine how subscriptions are evaluated and what information goes into the notifications.

3.3 User Organization

- Clubs on campus, each having their own heads, organise various events throughout the year.
- There are event managers for every event whose job spans right from campaigning, making announcements, making posters, etc. to the actual organization on the day of the event.
- The students are informed about the events through Facebook, WhatsApp, class announcements, etc.
- The students are informed about the time and venue of the event through the same means and any queries are clarified by contacting the event managers directly.
- Students who contested in the events, obtain results or pictures of the event through Facebook pages or they are individually contacted.
- When it comes the management of books on campus, students who know their seniors, try to obtain some required textbooks.

4. System Objectives

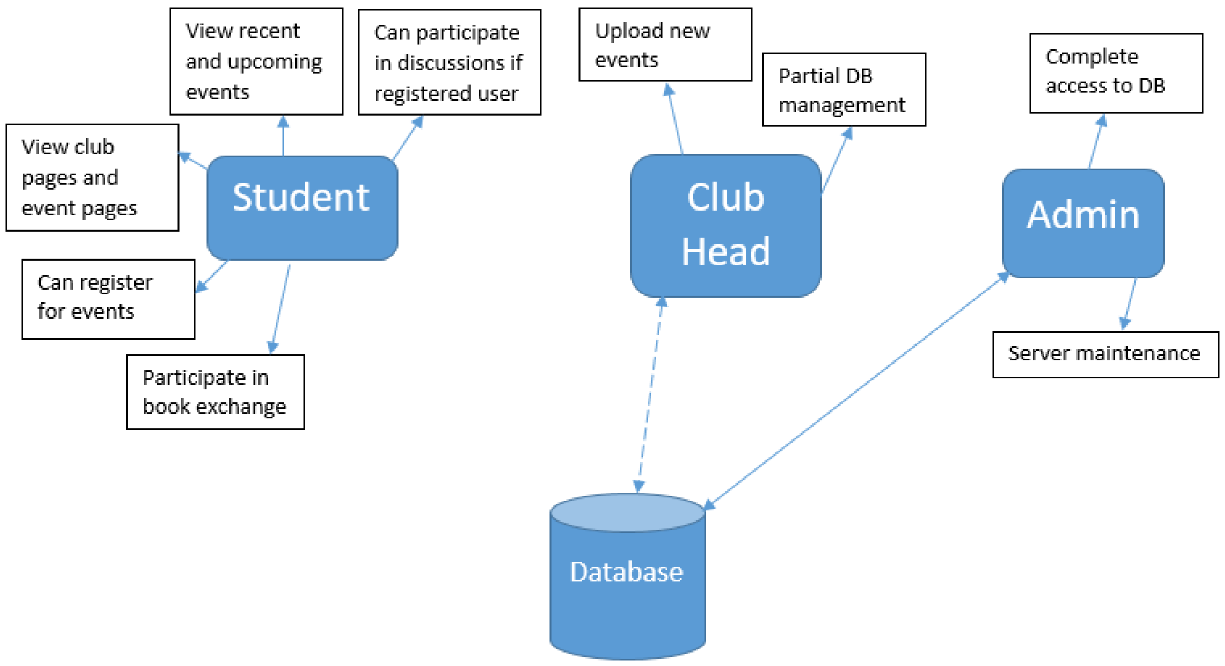
PES Times is projected to be a website, with one page dedicated to each society/club to post their upcoming events and other activities. The main objective is to provide a common social ground for interaction and event promotion with proper interfacing between clubs and students. The website would be self-sufficient with one admin for each club to manage (update & delete) club content and supervise registration entries for their events. No manual activity from developers end would be required after product completion.
To access all the features, students will have to sign up initially after which they can login from their accounts. The registration process for upcoming events would just require a student to sign in and click on ‘Register’ button, thus, preventing the mundane task of entering same basic entries (like name, USN, contact info) again and again for each event. Students with an account would be provided with regular push notifications for their registered events. Students would also be able to communicate via discussion forums under each event where they can put across any queries, suggestions and experiences.
A book exchange forum would consist of students entering book details in either ‘Needed’ or ’Offered’ section, based on which a search mechanism would find a suitable match from pre-existing entries. The development is likely to span over a period of 7-8 weeks with 5 weeks for development of both front

and back end of different sections and the rest for collaborating them together along with some final touchups.
Thus, the system would not only inflate the rate of participation by students in college activities but would also serve as a general platform for partaking resources and interaction.

4.1 Description of Products and Services

- PES Times is an online web based portal that will serve as a common platform for all societies/clubs that are part of the college, to post and advertise their activities.
- In present scenario, there is no interface between students and societies/clubs in the college.
- The only means of spreading a word include obsolete PR approach of posters and class announcements.
- By this project, we expect to give each club web presence to post their content thus adding a new dimension to event promotions.
- The club content can vary from upcoming events, objectives of the club to its history and recruitment criteria thus, providing an informative guide.
- It will also serve as a forum for discussion among students regarding events in the college.
- The Book exchange forum would not only endorse partaking but also will promote interaction between students from different batches.
- The ease of event lookup and registrations, offered by this project, is bound to bolster student participation in activities.
- PES Times would not include any notifications/registrations for events organized by departments.
- It would also not handle any money transactions if required for any of the event registrations or book exchange deals.
- Only those students having an account would be able to access Book exchange and Event registration features.
- Content posted by clubs would be visible to any user visiting the site.

4.2 High level Block diagram showing the solution


View recent and upcoming events
Can participate in discussions if registered user
Upload new events
Partial DB management
Complete access to DB
View club pages and event pages
Student
Club Head
Admin
Can register for events
Server maintenance
Participate in book exchange
Database

4.3 Targeted Customers and Benefits

The college life of a student is like enjoying the calm waters before diving into the treacherous oceans. He learns, he practices and he evolves so that he can face the merciless competition of outside. One of the key factors for that evolution is the participation in various kinds of events, which tries to bring the rules of outside world inside the walls of the colleges.
Our website is specifically dedicated to the students of a particular college (in this case: PES University). As already mentioned earlier, the birth of this idea came from the fact that there was no centralized portal where all the clubs can put their event details for students to participate.
So this website in a very loose manner can be said as a website for the students, by the students and of the students.
As per the conclusion from the survey, most of the students who are the day boarders depend upon the information and updates provided by the hostel students. A tradition followed very rigorously, but still very ineffective. Not every student knows everything, and even if he/she knows it might not be what the other person is looking for. So to save the interested pupil from this endless loop, the website seemed to be the way out.
The content of the websites will range from the upcoming hackathons, to the volunteering work and covering every cultural and educational aspect of the respective clubs. The website will be well managed and displayed. For example, if a student wants to browse through the upcoming events according to the club or by the kind of the event, will have full control of the way he wants to do it.
In addition to that, the website will have a separate corner for study related things. There will be a feature for book exchanges between students of the college. The students can put up the names of the books up for sell and the interested buyers can send the request to the concerned seller. Internally

database will create a list of these buyers as per the first come first serve basis. The student at the top of the list will be able to see the contact details of the student who is selling the book and vice-versa.
Most of the times in typical student's inbox, there is also high possibility of a good opportunity in a student's career like workshops, talks on higher studies as well as internship opportunities getting vanished. This website will keep track of all such opportunities and places it very systematically on the website.
In conclusion, this website is basically a digitization of the happenings around campus and it is open to all students.

4.4 Technology Considerations

As such there are no specific hardware requirements for the users. This service can be utilized using any device which can be used to access the Internet.
Software requirements are:

- Web Interface: a network that supports the HTTP/HTTPS protocol will be used.
- Server hosting: a JSP (Java Server Pages) servlet will be hosted by an Apache Tomcat web server.
- Database: PostgreSQL database
- Server- Database communication : Django
- Front End: HTML, CSS and Bootstrap
- Language for algorithm implementation: JavaScript

5. Product/Service Marketplace

A portal or platform to keep the students updated about college events and encourage interaction among them is an obligation for any college. It is a contemporary event promotion instrument to spread word across more effectually. The existing portal only has the transcripts and courses registered by every student. There also exist some apps but perhaps due to lack of marketing, these have not flourished well. Other colleges already have a platform in place for college events and registrations. By providing a more coherent way of sharing college event details and registrations for the same on a single platform we can expect participation rate to augment leaps and bounds.
The product is targeted to students of PES University. It is open and accessible to all but features can be explicitly used by students with an account.
The portal will allow various clubs to directly update their space in the platform. This will allow students to get the latest information without resorting to complex ways. It adds a new dimension to campaigning thus making the job of club members cushy, yet effective. Tedious form completions at the time of registration are reduced to a button click Vis a Vis internal database linking for logged in users.

6. Marketing Strategy

PES Times differentiates itself from its competitors such as other college related websites by allowing the students to find out about the latest news and events at PES and register for it all at one place. Instead of finding out about new events through disruptive promulgations and misleading posters, that hardly captivate any interest, the PES Times is an appealing alternative.
The portal allows the students to register for any upcoming event on the site, independent of the club involved. Registered pupil would be provided push notifications regarding their upcoming events to their email-ids entered at the time of signup.

Some marketing methods to be implied on the product:

- *Creating a Facebook page:* PES Times will have a dedicated Facebook page where recent exploits of the product would be broadcasted.
- *Promoting its Uniqueness and Exclusivity:* This portal is one of its kind and is exclusively dedicated to the students of PES. There will be no teachers on the website, making it a free environment to express one's views.
- *Spreading a Word:* Approaching club heads with the perks of this product will be one of the most effective marketing strategies. The heads will inform the members of the club who will in turn inform the other members of the college.
  Word of mouth especially in a college campus is after all the fastest way of promulgation.
- *Captivating UI & Accessible features:* An appealing UI coupled with serviceable features is the perfect recipe for a successful product.

Thus, we plan to use contemporary materialistic templates to design the interface. The portal will also have exquisite features like a book exchange facility that will allow students to partake their books of the past semester with other students that need it for their current semester.

7. Schedule

It is important to plan a schedule and stick to it through the course of project development. The following are the dates with the significant stages of the project that need to be completed.

19 Jan 2016:	Project topic selection and discussion with mentor
23 Jan 2016:	Begin development of feasibility study
29 Jan 2016:	Design of framework
31 Jan 2016:	Submission of final draft of Feasibility Report to the evaluating team
3 Feb 2016:	Work division and beginning of product development
13 Mar 2016:	Completion of work assigned to each member
6 Apr 2016:	Developed product by combining the work of all members
7 Apr 2016:	Beginning of testing and debugging
17 Apr 2016:	Final product, ready for presentation

8. Issues

Following are a few issues that we may encounter during the course of the project:

- Creating separate interfaces for club admins and normal users
- Since the approach is a client-server model, the main issue remains the maintenance of the server. If the server crashes, the entire operation comes to a standstill.
- Authentication of new users is a big issue until unless a legit database of all USNs is obtained from college.
- No inclusion of any inter-college events or ones organized by departments.
- Difficult to implement queuing feature of book seekers in book exchange section
- Only one admin for one club is also an irrevocable issue. There may be different sections in big clubs but all will have to upload through one admin.
- Implementing push notifications along with facebook sharing is going to be tedious.
- No registrations for those events that require a fee for registration.

- Convincing club admins to use the product and upload their content on it is a big task.
- The portal is solely meant for students of PES. Students of other colleges will only have access to limited features of the website.
- Scalability of the website, since the number of students increase significantly every year.

9. Assumptions and Constraints

Following are the assumptions:

- Each user will be having a unique standard ID.
- Centralized Database on a dedicated server.
- No money exchange involved during book exchange and registrations.
- All clubs are headed by a single individual.

Following are the constraints:

- Only those students having an account would be able to access Book exchange and one click registration features.
- Only one club-admin to manage and update club content.
- No money transactions involved during book exchange would be entertained by the product.
- Product is dedicated to the students of a particular college where intercollegiate events wouldn't be visible.
- PES Times would not include any notifications/registrations for events organized by departments.

10.Alternatives

The following is an alternative functionality for the web portal:-

❖ The portal can have a section dedicated to academics. This section will include virtual classrooms. Additionally, it will have discussion forums for doubt clarifications. Teachers will be actively involved in these discussions. Following the lectures, assignments and relevant material will be uploaded on the portal. Through this feature, the student can track and manage his course work as well as record his backlog.

The primary reason for discarding this alternative feature was its conflict with our idea of the product being a social platform for students for cultural exchange, along with the complexity of its development. For the given amount of time, it is seemingly not feasible. The main intention remains, having a working product by the end of the course.

11.Findings and Recommendations

This is a portal first of its kind keeping students updated about everything that is hot and happening in college. Targeted customers will be the college students themselves.

The following are the implementation details discovered through the study:

- A powerful backend built using Django to take care of generating news feeds and interact with the database

- PostgreSql for the RDBMS
- A website built using HTML5, bootstrap with the contemporary materialistic design
- A scalable website to accommodate increasing number of users.

The following are a few recommendations to make the product better:
- Developing a user friendly mobile interface along with the proposed website.
- Study corner should contain general study material related to the events currently happening. (For example, during a coding event we need pre-requisites related to algorithms, data structures .During a quiz event we need to read up about the topic on which the quiz is based on).
- All the clubs must be given equal importance and weightage.
- Restrict the number of admins per club who can access the database.
- Establishing a separate database for website activity thus maintaining a history of website activity.

These are a few pros of the proposed approach:
- Common platform for all the activities in college and thus it is easy to follow and keep track.
- Keeps the students updated about the latest events via the push notifications.
- Easy for the students to register with just the click of a button. No need of filling out endless forms.
- Increases overall participation as more students will be aware of the latest events.
- Separation is imposed between clubs so that we can maintain a record of all the activities of each club and the participation strength.
- Book exchange feature promotes partaking and interaction.

Here are the cons:
- Constant push notifications about events that may not be relevant to a user’s interests.
- Books feature may appear disjoint with the fun activities in college.
- Not accounting for events being organized by departments.

The product success rate should exceed standards since this portal is the first of its kind for PES. Looking at the convenience and thrill that the portal provides, it is evident that it will be appreciated and operated by every student.

11.1 Project Objectives

Following are some issues concerning development and implementation.

- Assumptions, Constraints and Limitations:
    - College will give us the adequate permissions to host our project on their server.
    - Students/Staff in the college make frequent use of our service.
    - Students/Staff keep periodically updating information about events happening and also post required data in the "books exchange corner".
    - Database is centralized.
    - All monetary issues are solved in person. Our project is not liable to any such issues that could appear.

- Results of research on hardware and software alternatives, technology, marketing, finance:
  - Technology:
    Backend is built using Django and PostgreSQL to fill the tech stack. Here are a few reasons for choosing Django:
    - *Django is time-tested*
    - *Django has been crowd-tested*
    - *Tons of Django packages available*
    - *Django has excellent documentation*
    - *The Django community is amazing and supportive*

    Having said this about Django, now it's time to let you know why PostgreSQL? Here are some key points to understand why we chose PostgreSQL.
    - PostgreSQL predicates are just ordinary expressions.
    - Ordinary SQL
    - Better support for JSON
    - Updatable and materialized views
  - Marketing:
    The whole motive behind PES Times is "Everything in one Place". Eg : All events now have a common interface. Many activities like book exchange, placement information broadcasting explains "Everything in one Place". Now let's try to understand why everything being kept at one place is such a big deal.
    - **Personal Level** - On the personal level, it saves me a lot of time/energy. Me as a student now, do not have to go looking for the admin or event head to know more about the event, or keep asking my seniors if they're ready to exchange books with me.
    - **Team level** - The event admins now do not need to break their heads thinking about the best way to reach out all the people at college. They do not have to create google forms for every new event that happens. Those factors can have a measurable effect on getting the project done faster.

    People would definitely prefer a system which is more efficient and one where things happen more systematically than something that doesn't meet those criteria. PES Times sets itself apart from its competitors by allowing the students to find out about the latest news and events at PES and register for it all at one place. Instead of finding out about new events through disruptive promulgations and misleading posters, that hardly captivate any interest, the PES Times is an appealing alternative.
  - Organizational:
    Once the project is finished and up on the server, there is so need for more staffing. Maybe minimal increases might be required to only monitor or do some analytics. There is no work involved in managing any new users or the authentication process to some extent.

Anyone who is identified as a student of the college, is given a USN and that is all he/she needs to use our service. Absolutely no need of other facilities or capital investments.

- Significant Risk Factors:
    - There is only one server and if it crashes for any unanticipated reason, the entire operation will come to a standstill.
    - In the current implementation of the project, there hasn't been any specific measures or use of methods like OAuth and OAuth2 which is used to take care of authentication. A simple Django-based authentication has been used.
    - There is no backup of the data that we might be storing whatsoever. Which means we aren't taking into consideration the database size.
    - Flexibility is not given to event admins. There can be only one event admin and it's only him or conceptually his account if anything exists which can make changes to their event.

- Feasibility Recommendations:
  Keeping aside all the above factors mentioned in the report like assumptions, constraints, risks and marketing strategy, this project intended to be accomplished within the given span seems definitely doable.

  To best implement our project, there are certain points that we have to keep in mind. The following recommendations will explain how we can make our work more feasible.

    - We should think more about building a product with more functionality than thinking more about how its appearance could affect the project once it's complete. Having a basic and simple UI with a completely functional project seems more appealing. So ensuring that the UI/UX is kept as simple as possible could count as one recommendation.
    - More discussions regarding the problem we are trying to solve is needed. Meeting the clubs, enquiring what they would want to have in a platform like PES Times if they were to use it would definitely help.

Project Plan

1. Deliverables of the Project :

The final product is a unified PES University web portal for the students and by the students. It is an information center for upcoming events to a social platform for the students to discuss and socialize, PES Times is an attempt at making all the scattered Facebook pages and groups for different clubs and events obsolete.

There will be features such as newsfeed. The web portal has a menu for recent events, upcoming events and all other events happening in the collage. There will be option for both registered user and anonymous user. Registered users can start a discussion, comment, set personal remainder and also share the events on other social networking sites. Anonymous user will just be having read privileges. The event organizer can set up an event page with event details and registration link. The web portal provides automatic input of details if user is logged in and pushes mail are also sent once the user is registered. There will also be a page for the college clubs and for incoming and outgoing events. Effective searching is also provided to the users that can give them option like search by club or search by domain. A book exchange corner is also provided to the user for exchanging, donating and reserving books.

The team who reviewed our feasibility report found the project idea innovative and found that it focuses on current issues of PESIT.

Some points were specified by them such as:

1. Plays and performances by college groups for event announcements and promotions have not been mentioned in the current systems.
2. Information regarding whether the system will be fully automated or it will include any manual activities hasn't been provided.
3. Info like means of communication between person offering the book and person receiving the book is not specified (like phone number or address).

We are going to consider the suggestions given by the other team and are going to implement them in the project.

Customers are expected to sign up the registration page with the help of their USN and password. Then they can login to register for the events. Customer can view the time, date and the venue for the event happening in the collage.

There are other website which are already used for event management and are successfully running but there is no specific website for the PES collage. Currently all the events pages are made in Facebook and specific website is made for each of this event. Thus making of this website will allow all the event organizer of the collage to post about their events in the single website made for the college itself.

2. Process Model which you intend to follow :

Software systems come and go through a series of passages that account for their inception, initial development, productive operation, upkeep, and retirement from one generation to another. It begins with background and definitions of traditional software life cycle models that dominate most textbook discussions and current software development practices. This is followed by a more comprehensive review and high level planning. The PES times project plan intends to follow a lightweight agile framework, scrum. Requirements would be divided into sectors and assigned to each scrum. Sprints are on weekly basis and scrums will be cross-functional.

Benefits of choosing the scrum methodology:

- In a nutshell this means that the development can start fast, but with the caveat that the project scope statement is "flexible" and not fully defined. It is a lightly controlled method which insists on frequent updating of the progress in work through regular meetings. Thus there is clear visibility of the project development.
- Daily meetings make it possible to measure individual productivity. This leads to the improvement in the productivity of each of the team members. Issues are identified well in advance through the daily meetings and hence can be resolved in speedily
- Due to short sprints (weekly) and constant feedback, it becomes easier to cope with the changes.
- It is easier to deliver a quality product in a scheduled time.

3. Identification of the upstream-downstream partners needed for the product :

Upstream:-

Products used in the development of the web portal:

- HTML, CSS
- Javascript
- Bootstrap
- Django

Downstream:-

The usage of this project is exclusive to the students of PES University. Among these users, the people involved with the clubs and events, for instance, the admins, are supposed to be in touch with the portal more directly. A more direct involvement comprises of being in contact with the developers of the website, and being active on the portal in order to update and create the events and post news related to the same.

4. Resources needed for the project/product :

Since the product is an application software, it has no requirements for hardware or cost as a resource. Effort and software are only resources to be worried about. All the frameworks and tools required for this product are open source and so availability of this resource is not an issue. Since time is of the essence, the resources that we really need to efficiently manage are time and effort. Some of the tools/frameworks required are:

- Django (python based web framework)
- Postgres library for python
- A browser (For running html-CSS and JS code)

Apart from this, we also need information about different clubs, their objective, history, members and other similar data. Effort is required for development but before that both time and effort is needed in getting the team familiar with new tools and mechanisms. Efficient resource management will be done using tight deadlines for learning and development phase.

5. How are you organizing your team in the project :

The Project is divided into sub-teams, each handling one of the mentioned postulates:-

- The home page for the portal
- The Club Pages
- Event registration
- Administrator interface
- Book Exchange

The reason behind this form of division was to keep minimal interdependencies between different sectors so that the product can be developed concurrently thus, utilizing time, effort and other resources efficiently.

In our development section, there is no need to freeze requirements and wait for the completion of a section.

Whatever interdependencies are there can be satisfied using dummy values to facilitate parallel creation of modules.

The admin interface needs the signup table database which will be a part of the home page section. The homepage needs the event table that will be a part event registration section and lastly the club pages needs the event table. All the dependencies are database related and can be compensated using dummy values.

The Book exchange feature is an add-on with no interdependency whatsoever. The work has been evenly divided into self-organizing teams, each with a deliverable and a time constraint, in accordance with the scrum framework for product development.

6. Standards-Guidelines-Procedures :

Development Guidelines: (Course of action)

The project began with narrowing down the problem statement. After assessing the feasibility of the idea, we framed certain requirements. An in-depth analysis of the initial requirements was done and we partitioned the project into non-repeating interdependent activities. Each activity has a sub-team working on it and has its own backlog and a separate module as deliverable. Integration of all the modules would be done towards the end. We plan to schedule timely meetings to monitor the progress and manage development time and other resources.

Development Standards: (Established Practices)

The product is following incremental and agile models of development.

- Feasibility Analysis - Taking into consideration the scope, scale and realization of the project in scheduled time, along with effort estimation
- Planning Requirements - Planning what frameworks, languages and designing tools that would be required.
- Defining Requirements - Django framework for server with HTML-CSS for user interface and Javascript for client end.
- System Design - Designing database with certain interdependencies between tables and drawing a rough layout for client interface.
- Implementation - Creation of separate modules by different teams and integrating them into one product.
- Testing - Testing various cases and boundary conditions.
- Deployment and Maintenance of the portal.

Procedures:

- Creating a main page having link to all hot events and club activities.
- Maintaining a database of students with accounts for one click registration.
- Making separate pages for clubs to post their contents.
- Designing an event management interface for creation of new events and a database having registered entries for the event.
- A book exchange interface with a database maintaining records of donors, recipient and transactions.
- Linking all the modules together in one portal.

7. Communication Mechanism :

Communication among team members proves to be very crucial as it ensures all the members are up to date with the strategies, changes in code and ideas. It is essential to have good platforms for conversation between team members so that ideas can be put across effectively.

For project-related communication among team members WhatsApp and Google groups have been created. Through these groups we can communicate ideas, carry out discussions and make important decisions.

For code sharing, regular team meetings and sub-team meetings are carried out. In addition, a GitHub repository will be created to upload the working code of each sub-team. Here, all the bits of code are combined. The repository will help us implement practices like pair programming for efficient deployment of the product.

8. Risks :

Following are few of the known risks:

- Not all club admins would prefer to login through our portal and promote their event. They might prefer to have their own website.

- In the book exchange section we aren't verifying the authenticity nor the condition of the books that have been put up for sale. Buyers may be misled into buying older versions or end up paying more than the book's resale value.
- From a user's perspective his news feed might get spammed with unwanted event updates and he might overlook some events or stop following the news feed.

Why we are willing to take these risk and stick to our plan:

Despite all these risks we strongly believe that our portal will be popular and successful in keeping the students informed about the intra college events and encourage more students to take part in them. Here are a few solid reasons to back it up:

- Process of registration made very simple and user-friendly.
- Students will not miss out on any event.
- Easy for the club admins to keep track of their past events
- Once few events start receiving a good response from the students, more clubs will definitely want to promote their events through our portal.
- Easy for the club admins to create a new event page. Since we've already designed a standard template they need not worry about it.
- Club admins can share their event page on Facebook and other social networking websites with just the click of a button.
- Club admins need not worry about building websites from scratch for every event.
- The book exchange section can be used to gain access to expensive books at pocket-friendly prices.

9. Quality Criteria :

The quality of a software product normally depends upon two things:

1. How well software is designed (quality of design)
2. How well software conforms to that design (quality of conformance)

(i) <u>Quality of Design</u>

Quality of Design depends mainly on two factors. The way the code is being written and the way the different code from the sub teams is integrated. The whole team has been divided into 4-5 groups and each group has a responsibility of creating one aspect of the website. So, this goes without saying that the code generated by all groups has to provide ease of maintenance, high readability and scalable. To achieve this, each group has been given a timeline by which they have to complete their given task. After which the whole team will sit down and merge everything. There will be some errors and some un-synchronization, which will form the next part of the project; to clear all the minor backlashes and generate the final product.

One advantage which our team will be exploiting a lot is that the backend and the front end of a given task are done by the same group instead of a separate group for backend and front-end. In the latter case we have seen that the time delay on the delivery of the product is more as compare to the former way which we have opted for.

Hence, the quality of design may not be perfect, but it will surely be on the standards of any given product.

(ii) Quality of conformance

Is the product as a whole ready for the consumer?

Are there any minor glitches in the product which may eventually crash the whole system?

Is the product fully compatible with the majority of the systems available?

The answers to these questions will form the basis for the second type. After the final integration of the product, a good quality of time will be spend on the various types of testing to see whether the product can withstand all the normal queries as well the tricky end cases.

With the normal procedure of doing all the different types of testing (unit testing, functional testing, performance testing), we will try to have a beta audience to test the product. Also while developing, each of us will keep in mind any loopholes coming into the picture and looking for a way to overcome it.

For the final presentation, each and every module along with each and every feature will be checked and performance documented, thus completing the *quality of conformance.*

10. Work Packages :

We have divided the project into 5 domains. Each domain has its own back-end and front-end part. After the completion of all domain work, we have to integrate those domains into a single domain by removing the redundancies. Five domains are as follows.

a. Book exchange: This domain is related to study related things. Student who wish to sell the books can put up the name of the books and student who wish to buy the book can request the seller. Request is served on First come first serve basis. There will be internal database designed to handle these.

b. Admin Interface: Interface is accessible to admin who is head of the respective club. Page consists of three parts, database, create event and promote. Admin can create new event as well as can share it in social networking sites. Admin can also update the club page by removing or adding information related to the club by maintaining the database.

c. Main Page(news feed): This page contains information about events like ongoing events, upcoming events and recent events. Page displays all the clubs in PESIT. Once student sign up, they can log in and use the website. Admins can login separately from this page.

d. Event Page: Page displays all the details of an event selected by user. It also supports the user to register to an event via registration link.

e. Club Page: Page has all the information about the club (objective, members, history, gallery, etc.), ongoing events, upcoming events and recent events organized by club. Page is maintained by head of the club who has separate interface called admin interface to update the club page.

11. Budget and Schedule :

High level schedule:

Date	**Action to be completed**
19 Jan 2016	Project topic selection and discussion with mentor
23 Jan 2016	Begin development of feasibility study
29 Jan 2016	Design of framework
31 Jan 2016	Submission of final draft of Feasibility Report to the evaluating team
3 Feb 2016	Work division and beginning of product development
13 Mar 2016	Completion of work assigned to each member
6 Apr 2016	Developed product by combining the work of all members
7 Apr 2016	Beginning of testing and debugging
17 Apr 2016	Final product, ready for presentation

Detailed work breakdown with individual ownership:

Anirudh Agarwal	Admin interface: Back-end development Reports and reviewing Planning and database design for Admin interface
Mohit Mayank	Main page(News Feed): Back-end development Reports and reviewing
Manisha Rachel Dawson	Event page: Back-end development
Nagasundar	Book exchange: Front-end design
Navneet Singh	Admin interface: Front-end design
Neha M Kalibhat	Main page(News Feed): Back-end development Meetings, deadlines, management of groups and docs Reports and reviewing
Niket Raj	Club page: Back-end development

Parikshit Maheshwari	Admin interface: Back-end development Reports and reviewing Planning and database design for Admin interface
Prafful Um	Book exchange: Back-end development Reports and reviewing Planning and database design for Book exchange
Rohan Ds	Book exchange: Back-end development Planning and database design for Book exchange
Romasha Suman	Main page(News Feed): Front-end design
Sharath Kp	Club page: Front-end design
Smitha	Event page: Front-end design

12. Delivery means :

The working and tested product will be deployed on a server and when it goes live, it can be accessible by any student. The links to the website are shared extensively through social networking sites, WhatsApp groups etc.

Club heads may approach and obtain admin logins from the developers. Once admins have been identified they can go ahead and upload upcoming events and share the same on social networking websites.

Software Requirement Specification

1. Introduction

1.1 Purpose

The intended audience for this SRS includes all of the stakeholders in the PES Times project. The document will be used by team developing the project as the specification from which to implement the working program code. The document will also be used by the development as a statement of what functionality will be delivered during the project. The document will be used as a deliverable for the academic portion of the Software Engineering class and graded for its quality.

1.2 Scope

The PES Times is a web based portal that will keep the students of PES college abreast of the upcoming events of college in a convenient and consolidated manner. The portal will allow the students to track all upcoming events being organised by various clubs and keep track of their participation. It will also have a feature that will allow for book exchange among the students of PES. It will not be available as an application on mobile systems.

1.3 Definitions, Acronyms, and Abbreviations

PES: People's Education Society

Newsfeed: The continuous stream of upcoming events displayed when a user logs in.

2. General Description

2.1 Product Perspective

PES Times is a web portal made solely for the students of PES to track and manage ongoing events on campus. It aims to replace the current scenario of sparse groups on social networking sites and other websites of various clubs. The system is designed to have four main roles around the portal as shown in the diagram. The interactions between the portal and the user as illustrated as follows.

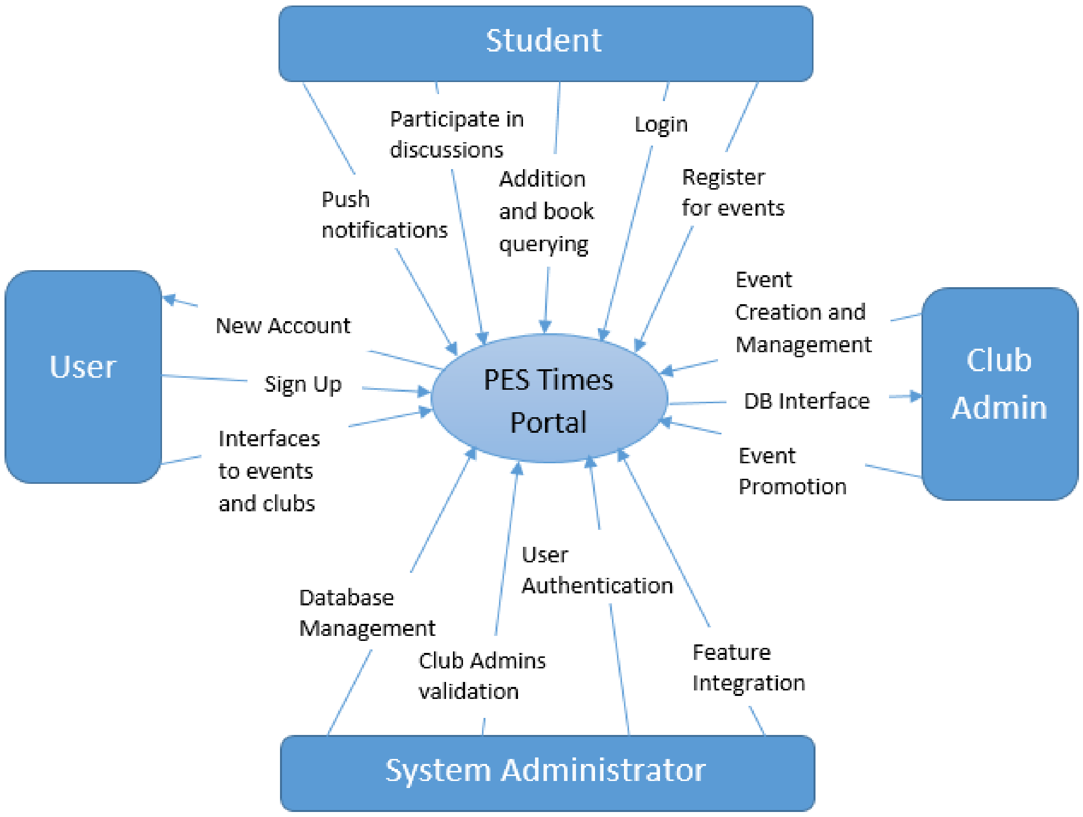


Fig 1: Context diagram for PES portal

It is important to note that Students and Club Admins have the basic interfaces to clubs and events. Students have some additional benefits in comparison to users without account as specified and Club Admins have a separate interface to perform event management. User authentication is done only to the extent of identifying and verifying the users and their passwords if they have accounts in the website.

2.2 Product Features

FE-1: One-click event registration

FE-2: Multiple clubs interfacing

FE-3: Sign-up for new users

FE-4: Log-in for existing users

FE-5: Event creation and management

FE-6: Event interfacing

FE-7: Discussion forum

FE-8: Club info-graphic

FE-8: Event Promotion

FE-9: Push notifications

FE-10: Registration database management

FE-11: Book-exchange database management

FE-12: Book-addition and query

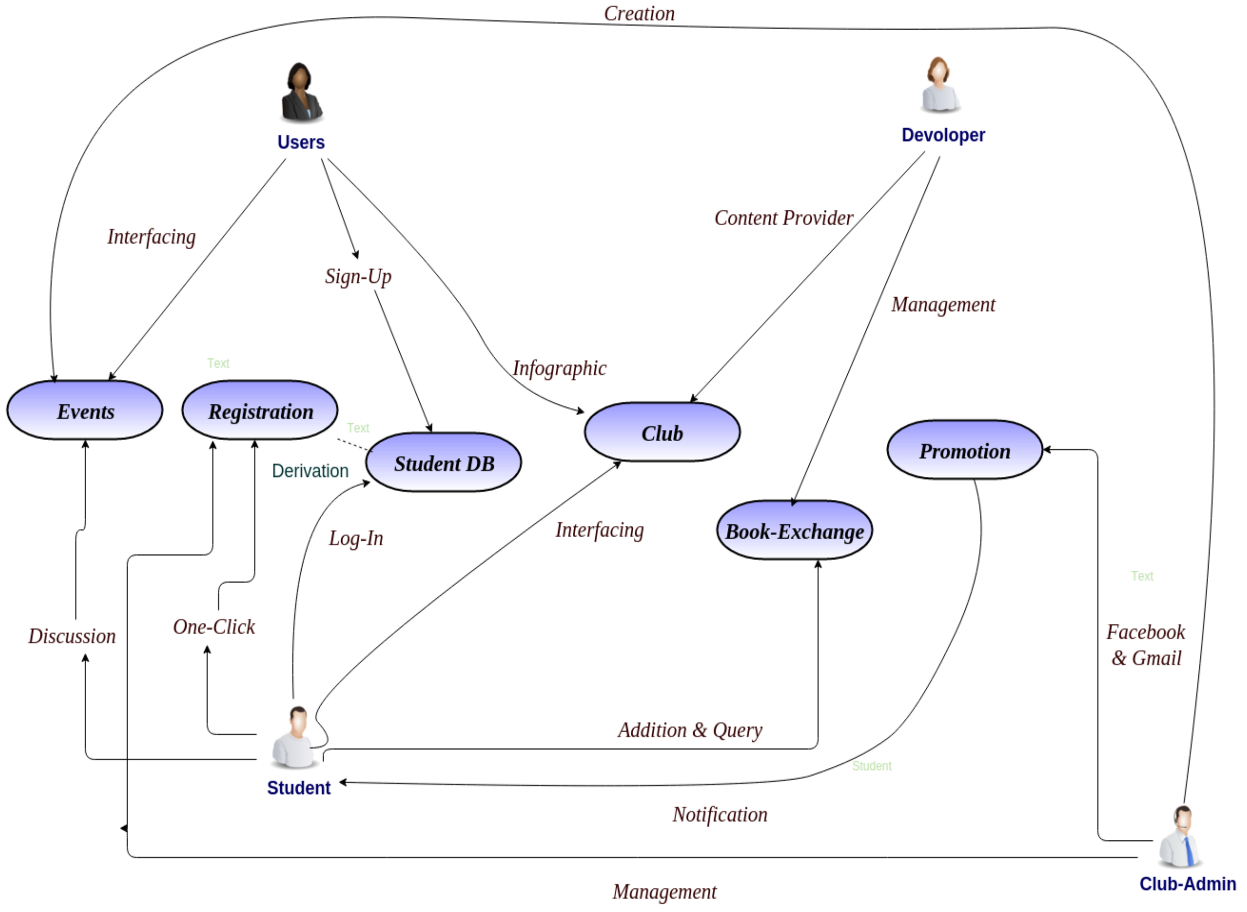


Fig:2 Relationship between features

2.3 User Characteristics

College Student	Student of the particular college for which the website has been developed will use it for the following purposes. ● To check the various events organized by different clubs ● They need to log in to avail the services ● They have the options to browse events under the name of the club or category ● If they want to participate in any one them, they will be able to register through the registration link provided ● A logged in student can also start a discussions by commenting under that event which he/she attended or wishes to attend. ● They can put up the name of the books they want to sell on the website and can look for the book names they want to purchase from the fellow other students as well.
User	The students from different colleges or any user without a login can view the websites but cannot register for the events or participate in any forum anywhere on the webpage. Logins are given only to students belonging to PES.
Club Administrator	Club administrator is a special student who, along with the above privileges will also be the head of one of the many clubs of the college. He would be given special privileges like creating, updating and deleting an event. When a club admin logs in, his username will be verified as the admin with the already present information on the database. This will enable the admin to access the administrator page of his registered club and their activities inside it.
System Administrator	System administrator will keep a track on the working of the website and is responsible to integrate any new features on the same. Also, if a new club emerges and needs to be put on the web page, the club administrator has to contact the system administrator to do the same.

2.4 Operating Environment

OE-1:	The system is not dependent on geographical areas.

OE-2:	The system shall operate in the newest versions of all web browsers.
OE-3:	The logged in user (same college student) can access all the features as compared to the students from different colleges.
OE-4:	Data is generated from the registration link (online forms) as well as the information generating during the booking for a book to buy or a book to sell.
OE-5:	Continuous service is preferred, but as long as there is no data loss, minor interruptions can be tolerated.
OE-6:	Javascript, Bootstrap, HTML/CSS is used for generating the user interface for the system.
OE-7:	Django Framework is used for creation of databases to store the information from the registration links and book exchange features.
OE-8:	To allow the students to fill up the required information to participate in events online Google Forms will be generated with the information fields provided by the club admins.

2.5 General Constraints

The PES Times portal will not be able to operate if there is no connection to the server. It needs a good framework without which it cannot interact with the database. The logged in user should be able to access all the features as compared to users from different colleges. Continuous service is required. Data loss and interruptions limit the operating environment.

Another limiting constraint will be its availability as an application on mobile phones.

2.6 Assumptions and Dependencies

Assumptions:

- Will be used by the students of the college. i.e. the primary target audience
- Students and Club Admins should have the basic interfaces to clubs and events.

- The PES Times portal should be able to operate on the newest versions of all the web browsers.
- A database is one another such assumption needed to manage and store the number of event registrations, to retrieve and add events etc. The PES Times portal uses Django as the basic framework that interacts with the MySql database.
- The system administrator will keep track of the website and integrate new features into the website as and when needed. Ex: If another feature besides the book exchange feature needs to be added.
- Club administrators should be able to access the page of his/her club along with the activities in it.

Dependencies:

- The users depend on the clubs and the newsfeed for information, the events for interfacing and the database for registration.
- The developers depend on the software tools being used i.e., Javascript, html, CSS, Django and bootstrap.
- The club administrators depend on the new events they can create and the promotions of such events and the students who will register for such events.

3. Specific Requirements

3.1 Functional Requirements (FR)

- What the system does:
    1. Welcome page with news feed of events and clubs
    2. User authentication needs to be done for Login and Signup.
    3. Keep track and display all the events – past, present and future.
    4. Separate Club interfaces
    5. Facility to allow the users with an account to register for events.
    6. Notifications for events.
    7. Admin specific functions:
        - Creation of events
        - Managing the events
        - Event promotion on social platforms and push mail
        - Updating the details of events and gallery for the clubs/events
    8. Book exchange: provision for students to exchange books among themselves.

- What the system doesn’t do:
    - If an event requires a monetary transaction, the student needs to submit the fees in person – the club doesn’t have support for online transaction.
    - Similarly for book exchange, online transaction facility is not provided.

3.2 Non-Functional Requirements (NFR)

1. Scalability – non-scalable, in general, because the usage of the portal is limited to PES
2. Availability – a server is required to be up 24x7
3. Recoverability – a backup has to be maintained of all the data up and running on the server
4. Security – basic USN and password login, no more security provided
5. Interoperability – interdependency between databases, e.g.: database for events has club ID (PK from club database)
6. Data Integrity – normalized databases ensure the integrity of data
7. Reliability – handled by the admins of the different clubs, so discrepancy might creep in due to some miscreants
8. Usability – a simple UI makes the portable usable by anyone and everyone
9. Environmental – web-based application

3.3 External Interface Requirements

- User Interface:
    - The Web application developed will be mobile friendly and enticing with the latest materialistic design.
    - Users can navigate between Newsfeed, book exchange section, specific club activities page via the navigation drawer
    - In addition to the navigation tabs present at the top of the page.
    - Users will be redirected to the events registration page on clicking on a particular event on the news feed. Users will be able to register themselves for a particular event with just the click of a button
- Software Interface:
    - **Database** - The system shall communicate with a database for the following operations
        - To retrieve the most latest and upcoming events from the list of events to populate the news feed.
        - To manage event registrations - stats about the no of registrations etc.
        - To allow the club admins to create and add new events to the list of events.
    - **AJAX** - The system shall use the AJAX concept for the following
        - Load images of past events only when the user demands it.
        - To fetch more events/activities from the database when the user scrolls to the end of the newsfeed and then extend the feed with the asynchronously fetched data.
        - To retrieve and display the list of sellers when a user searches for a particular book in book exchange section.
    - **Django based (for backend management)**
        - To manage the entire backend and the interaction with the database.
        - Processing the response to the queries made by the user and converting it to the standard JSON string format before sending it back to the client.
        - Read the data from the database, filter it and order it.

        - Django ORM: allows a developer to write Python code instead of SQL to create, read, update and delete data and schemas in their database - can speed up web application development, especially at the beginning of a project.
    - **Twitter Bootstrap**
        - To make the webpages highly responsive and furnish them with materialistic styling/design.

Architecture and Detailed Design Specification

1. Architectural and component-level design

The main architecture of the Pes Portal consists of Django-based web apps that render different pages in the website; Newsfeed, Clubs, Events, Event admin interface, Book exchange forum. These apps are built in the Django framework using different software and languages to perform styling, back-end implementation and database querying and connectivity.

The following are the designs of software components of the project while Django provides the framework.

- **Django -** Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source. The main features of Django are it is fast, secure and scalable.
  So, Django forms the nucleus of our project. It stores our mark-ups, stylesheets and JavaScript files. It decides the mapping between URLs and decides the behavior of the project. It also takes care of authentication, session control and provides a seamless backend functionality.

- **HTML -** Hypertext Markup Language, commonly referred to as HTML, is the standard Markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and web applications. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

- **PostgreSQL -** PL/pgSQL (Procedural Language/PostgreSQL) is a procedural programming language supported by the PostgreSQL RDBMS. It implements some ISO SQL/PSM features, like overloading of SQL-invoked functions and procedures. PL/pgSQL, as a fully featured programming language, allows much more procedural control than SQL, including the ability to use loops and other control structures. SQL statements and triggers can call functions created in the PL/pgSQL language. In our project Django querysets are internally converted to postgreSQL queries to make and query database tables.

- **Bootstrap –** It is a free and open-source collection of tools for creating websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It aims to ease the development of dynamic websites and web applications making them more appealing. Bootstrap is a front end web framework, that is, an interface for the user, unlike the server-side code which resides on the "back end" or server. We use it in our projects to enhance the look of our webpages.

- **JavaScript -** JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complementary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform. It is implemented to in our html files to derive the content dynamically from database.

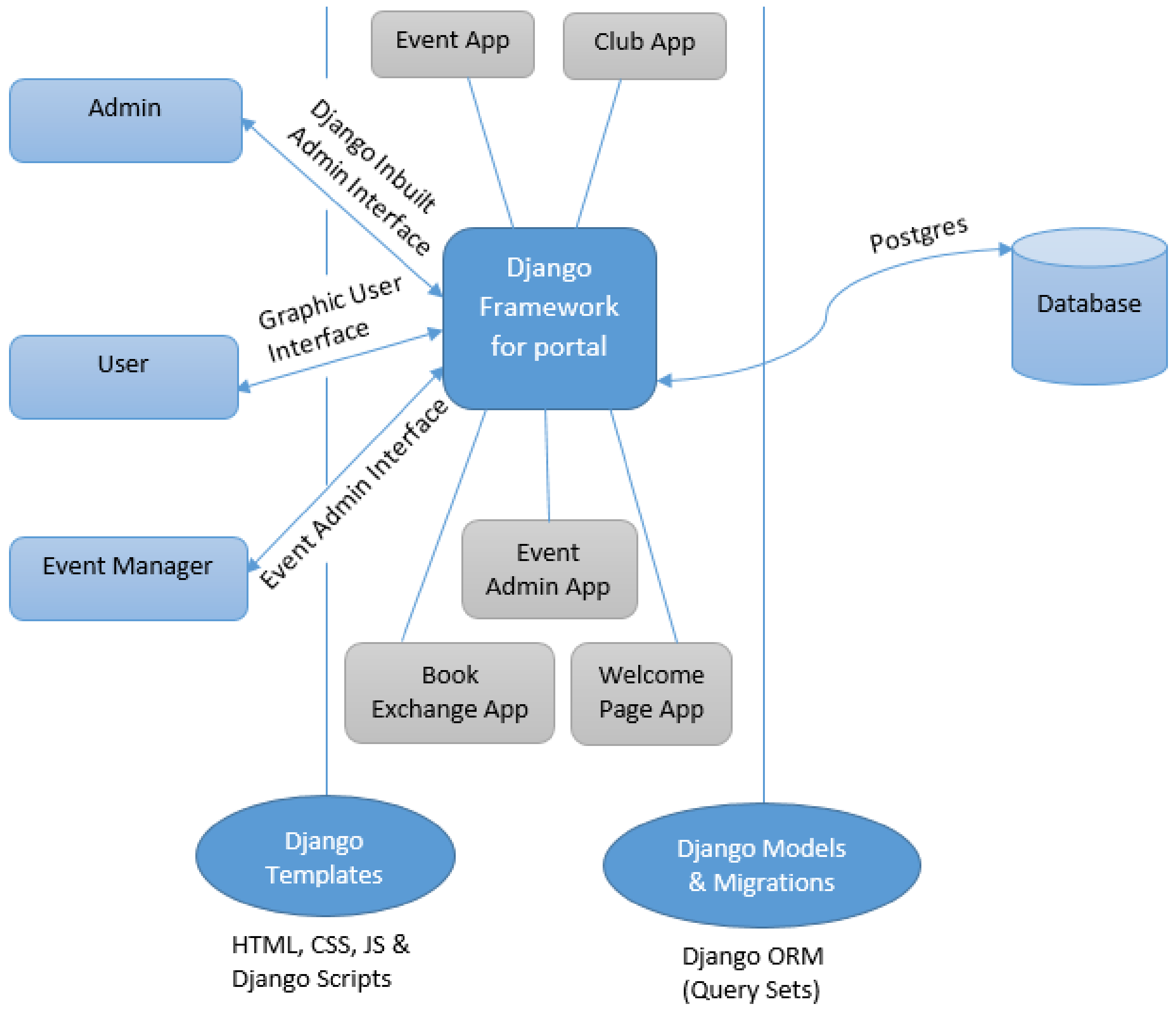
- **CSS** - Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.

So these are the main building blocks of our Project. The user on entering the particular URL will be rendered with a markup and this mechanism is controlled by the nucleus, Django. Django internally uses database engines, in our case, Django uses an engine called psycopg2 and provides an API to perform the basic operations with respect to Database. The markups have been styled pretty well using some open source tools like Bootstrap by which we can produce responsive and visually appealing designs.

2. System Structure

2.1 Architecture diagram

Diagram of all the interactions between the software components



2.2 Description for Components

Django, HTML, PostgreSQL, etc. And how they interact. Check the Architecture diagram and describe each entity in it.

Software components involved are:

A. Django:

Django is a high-level and open-source web framework, written in Python.

a) Django Templates:

Being a web framework, Django needs a convenient way to generate HTML dynamically. The most common approach relies on templates. A template contains the static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted, which get replaced with values when the template is evaluated, and tags, which control the logic of the template.

b) Django Apps:

A Django App describes a Python package that provides some set of features. A single project can have multiple independent as well as inter-dependent Django apps, each app having its own static files, templates, URL redirection, views and models.

c) Django ORM:

The ORM is an incredibly powerful database tool. It handles creation of your database, as well as insert, update, delete queries and some quite advanced querying using simple querysets. It supports multiple databases - MySQL, PostgreSQL, Oracle & SQLite are all supported out-of-the-box assuming you have the relevant Python libraries installed. In our case we use internal mapping to PostgresSQL.

B. HTML: Django Template Language

To make dynamic pages you need HTML which can populate the webpage with the data that you send from your python functions (handlers). It stands for Hypertext Markup Language, a standardized system for tagging text files to achieve font, colour, graphic, and hyperlink effects on World Wide Web pages.HTML consists of a series of short codes typed into a text-file by the site author — these are the tags. The text is then saved as an html file, and viewed through a browser.

HTML also has various software components like:-

i. JavaScript: Basic special effects and interaction is provided by JavaScript. It is used to program the behavior of web pages. It allows you to create highly responsive interfaces that improve the user experience and provide dynamic functionality, without having to wait for the server to react and show another page. It can load content into the document if and when the user needs it, without reloading the entire page.

ii. CSS: to control how your pages are presented, and make pages more accessible and also to define styles for your documents, including the design, layout and variations in display for different devices and screen sizes. You can place your CSS in the <head> of a document with an embedded style sheet, or attach a separate file that defines your styles with an external style sheet. To link an external style sheet to your document, you'll simply add a link to the style sheet in the <head> of the document. You can keep the styles separate from your HTML content which Helps avoid duplication, makes maintenance easier and allows you to make a site-wide change in one place.

C. PostgreSQL: Database Server

It is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards-compliance. As a database server, its primary function is to store data securely, supporting best practices, and to allow for retrieval at the request of other software applications. It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

In PostgreSQL, a schema holds all objects (with the exception of roles and tablespaces). Schemas effectively act like namespaces, allowing objects of the same name to co-exist in the same database.

Interaction between Software components:

Interaction between various Software components is obtained through Django. It interacts with each software components individually as you can see in the Architecture diagram above. For interacting with each software components Django uses different Python Modules:

I. Manage module as manage.py

II. Admin module as admin.py

III. URL module as urls.py

IV. Views module as views.py

V. Models module as models.py

The models are:

- Signup
- Club
- Event
- Registrations
- Comments
- Pending Transactions
- Seller

VI. Forms module as forms.py

VII. Setting module as settings.py

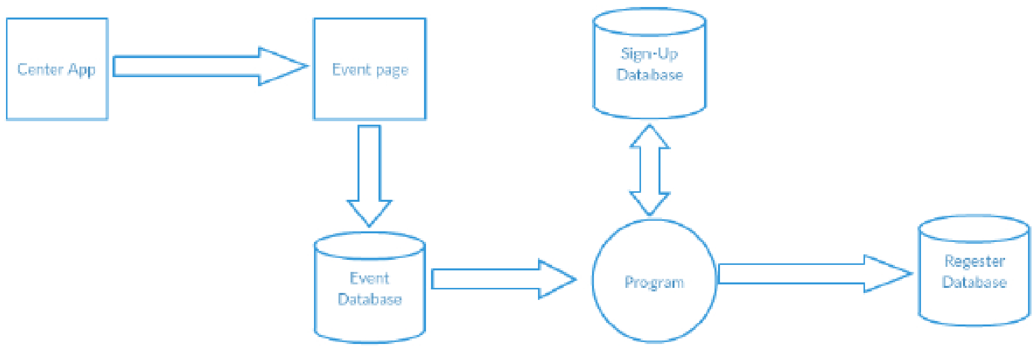
manage.py module is automatically created in each Django project. It is like a controller of the system, it steers everything. For a web framework, this means handling requests and responses, setting up database connections and loading add-ons. For this, Django reads a settings Module so that it knows what to load and set up.

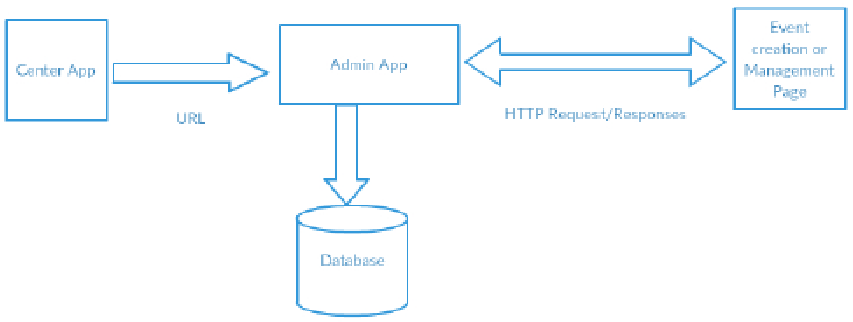
The entry point to Django applications are URLs. When a user accesses a URL of a Django-powered site or performs an action on an already loaded page of that site, The Django's URL dispatcher runs through the URL patterns in the urls.py module and chooses the first one that matches the URL requested by the user and passes it to a view for processing which invokes a Python callback function in views.py associated with the found pattern.

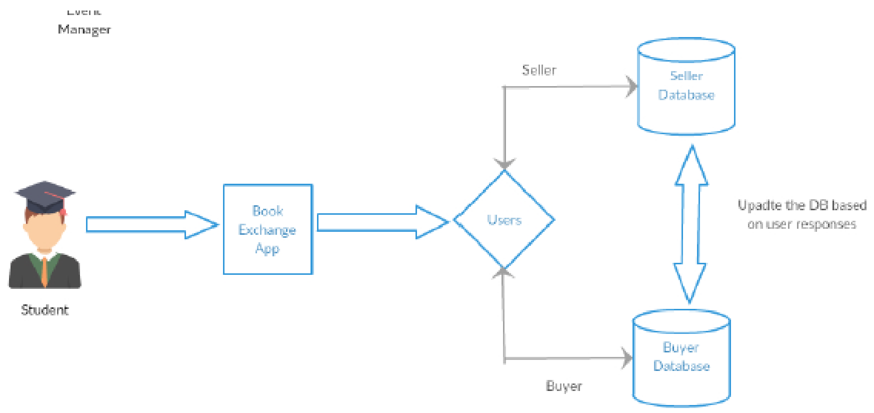
The view module is the user interface.URL requests are redirected here and from here can be further redirected to templates. Django splits this up in the actual HTML pages and the Python code (called views) that renders them. The view uses data models to obtain database data and then loads the specified template (an HTML page) and passes it the context including the obtained data mapped to template variable names. Finally, the view returns an HttpResponse object populated with the rendered template, or an HTTP exception if something went wrong.

The Model module in Django means the database plus the Python code that directly uses it. It implements Object Relational Mapping (ORM) and creates classes corresponding to tables in database. You capture whatever your website needs in database tables. Django helps you write Python classes (called models) that is used for updation and deletion of data in database. The database can be easily queried by instantiating classes defined inside models.py and using querysets with those objects.

2.3 <u>Interaction Diagrams</u>







2.4 <u>Describe usage scenarios and how you would test that</u>

<u>**Usage scenarios are:**</u>

- Website admin approves for a club. The club is added to the website.
- The club admin adding various events organized by the club.
- The club admin sending push mail to various email addresses as a notification for their registered events.
- Student registering for an upcoming event.
- Club-admin managing the list of registrations done for an event i.e., getting relevant details from registration list.
- Student querying for a book.

- Student uploading a book for the purpose of selling or partaking.
- Student checking out the details of a club and upcoming events

Testing:

- Dummy signups will be done both as a user and a privileged club-admin.
- These test club-admins would be given privileges for official club-pages with static content.
- Dummy events will be created and will be added to clubs.
- Deletion of events and registrations will be tested by these club-admins.
- The events are tested for the Facebook post and push mails by sending them to dummy accounts.
- Student signups, session maintenance after logging in for the event and the one-click registration` feature is tested for successful operation.
- Student uploading and querying for the book is tested by constantly checking the database for synchronous querying.
- The clubs, events and main page content would be tested by derivation from above created dummy data.

2.5 Architectural Styles and Patterns considered and for what reason

S.No	Architectural Styles/ Pattern	Intent of this pattern	Rationale for choosing or not choosing
1.	Client-Server	The system is divided into two applications; Client and Server. The client makes requests while the server is an always on entity that listens. The database is stored as procedures.	Being a website, the PES portal needs to be available consistently 24x7. This pattern is thus chosen to ensure that all client requests are handled at the server at all times. Another reason for choosing this system is the concept of thin client. The idea is to have least storage at the client as the database and most of the operations are confined to the server itself with just a light interface to interact with the client.
2.	Component-based	The application is decomposed into several reusable functional components with well-defined communication interfaces.	The project is divided into five major apps: Newsfeed, Event Admin Interface, Clubs, Events, Book Exchange. This pattern helps in establishing well-defined components. It not only helps in defining flow of data between components but also makes the product modular and user friendly rather than one

			big complex mess.
3.	Layered Architecture	Partitions the concerns of the application into stacked groups (layers)	This pattern is applicable in our project as the entire application can have three layers of abstraction; Database, Back-end and Front-end. The website is built through the above levels providing appropriate connection mechanisms between each pair of layers.

3.0 User interface design

MAIN PAGE:

Features the opening interface with newsfeed consisting of upcoming events. The bottom will have tiles representing different clubs each linked to their respective pages. The top will feature a navbar with links to Book exchange section, links to different sections of the main page along with link to event management section .It will also consist of login and signup buttons for new and registered users respectively. A search button will be provided to navigate through the entire site with ease.

CLUB PAGE:

This page has details about the clubs including a brief description, objective of the club, accomplishments, members, Facebook and Twitter links for the club and details about the club-admin.

It also presents the recent and upcoming events organised by that club each linking to respective event pages.

EVENT PAGE:

This page basically contains the event name and description of the event along with the time, date, no. of participants and the venue of the event.

It has the feature of one-click registration using which a logged in user can register for that event without filling any forms.

It contains a comment forum where people can discuss and comment about the event.

It also contains a gallery with images pertaining to clubs past endeavours.

BOOK EXCHANGE PAGE:

A search box is present to search based on author, book name, seller. Based on the query, the available books are displayed. The student selects the books he wants and waits to be contacted by the seller. The seller will have an interface with a list of potentials buyers from which he can contact any one.

The payment negotiation is managed between the buyer and seller in-person.

EVENT MANAGEMENT:

Here, a club admin who has special privileges can create new events, manage events as well as registration entries done for those events and promote via push notifications and on social media. Event Creation interface will require to admin to mention a brief description, date, time and venue of the event along with various other compulsory and optional fields. A poster for the event is uploaded and contact info is given. Event management interface provides admin with lists of events hosted by his/her club along with registrations done by the people for those events. He/she will have the freedom to delete any of these entries. The club admin will be given the privilege of sharing events on facebook, twitter and send push notifications to the email id’s of all the registered students using the promotion interface.

4. Detailed Design Approach

Top-down approach

- A top-down design approach starts by identifying the major components of the system, decomposing them into their lower-level components and iterating until the desired level of detail is achieved.
- Top-down design methods often result in some form of stepwise refinement.
- Starting from an abstract design, in each step the design is refined to a more concrete level, until we reach a level where no more refinement is needed and the design can be implemented directly.
- Starting from an abstract design, in each step the design is refined to a more concrete level, until we reach a level where no more refinement is needed and the design can be implemented directly

4.1 Design patterns considered and for what reason

S.No	Design Pattern	Intent of this pattern	Rationale for choosing or not choosing
1.	Creational	Deals with the best way to create instances of objects as the application should not depend on the method of object creation	This pattern is incorporated. The models created in Django are all instantiated to create databases following ORM(Object Relational Mapping).Even custom forms are created by instantiating classes . This creation in any case doesn’t affect or cause any changes to the behaviour of the website but it eases developer's job of form, database creation and querying.

2.	Structural	Deals with identifying a way to realize relationships between entities	This pattern is used in terms of the database models where relationships are identified with foreign keys between models. Different apps on the portal are related in terms of data interdependencies and linked tables in shared database.
3.	Behavioural	Concerned with identifying common interactions between objects and realizing them	This pattern is used as the apps of the website are required to interact with each other in terms of passing data across sessions.

Traceability Matrix

FR-Functional Requirement

Requirement	Design Phase	Implementation Phase
FR-1 : Welcome Page	Django views.py Django models: Event, Club, Signup Django template: newsfeed.html	Views.py -> render_newsfeed -> fetch events and clubs from tables -> newsfeed.html
FR-2 : Event Page	Django views.py Django models: Event Django template: event.html	Views.py -> render_event -> fetch info about queried event -> event.html
FR-3 : Club Page	Django views.py Django models: Club, Event Django template: club.html	Views.py -> render_club -> fetch info about queried club -> club.html
FR-4 : One-Click Registration	Django views.py Django models: Event, Registrations, Signup Django template: event.html	Views.py -> render_event -> event.html -> on click of register button -> insert new entry in registrations table
FR-5 : Push Notifications	Django views.py Django models: Event, Registrations, Signup Django template: click_prom.html	Views.py -> click_prom -> click_prom.html -> on click of promote button -> email sent to all registered participants
FR-6 : Admin Interface	Django views.py Django models: Event, Registrations Django template: main_admin.html	Views.py -> main_admin -> main_admin.html -> redirects to Event Creation, Promotion and Management interfaces

FR-7 : Book Exchange interface	Django views.py Django models: Pending Transactions, Seller Django template: up_for_sale.html	Views.py -> list_books_for_sale -> up_for_sale.html -> has search feature enabled with provision to upload books

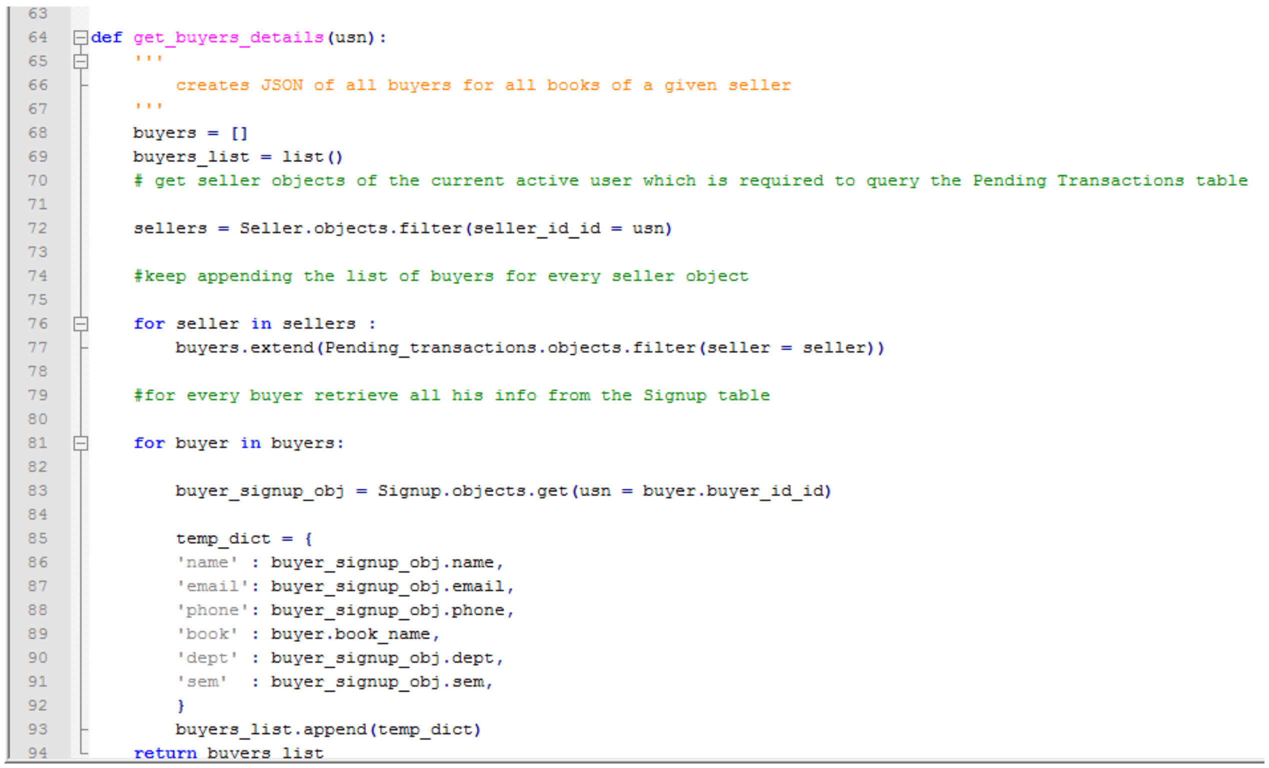
Implementation with Code

Our implementation now consists of 4 apps:

1. Welcome Page
    a. Renders the main welcome page with the newsfeed
    b. Contains a file base.html with html markup common to all the pages
    c. Contains functionalities for navbar
    d. Contains functionalities for Login and Signup and Reset password with email
    e. Renders event page
    f. Event page functionalities like One-Click registration and Comments section implemented in this app
2. Club Admin
    a. Renders pages for club admin interface.
    b. event creation, event management, view registrations and event promotion
    c. Only accessible to club admins. Sessions are used to detect this
    d. Implements the event promotion with Facebook and Twitter links
3. Club
    a. Renders page for selected club
    b. Displays the details for the club
    c. Displays the events of that club
    d. Gallery feature that is carousel of images of that club
4. Book Exchange
    a. Accessible only if logged in
    b. Displays all available books with sellers
    c. Once book is borrowed, the seller can log in and see the buyers
    d. Seller can select a particular buyer and remaining buyers are notified via email that the seller is not interested in lending the book
    e. Implements the upload book functionality

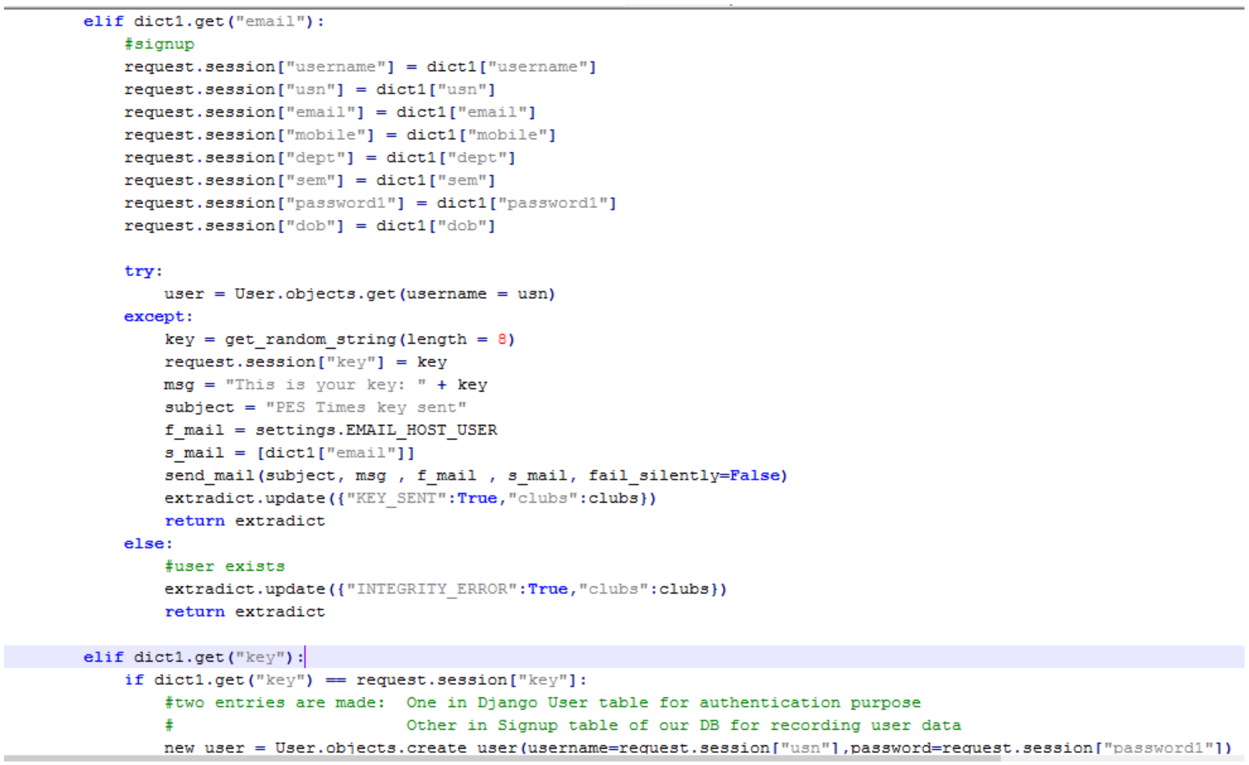
Following are code snippets of the important features of the product:

Book Exchange (Serialization of books):



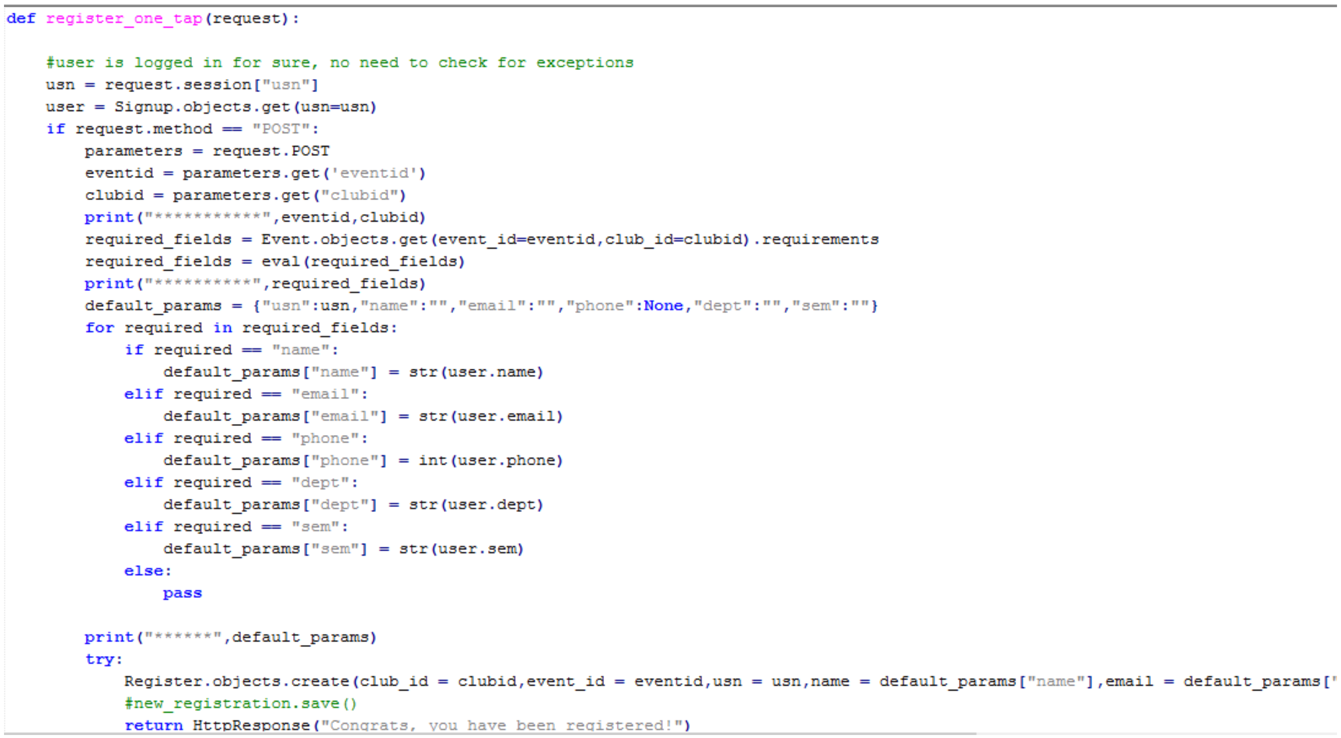
```python
def get_buyers_details(usn):
    '''
        creates JSON of all buyers for all books of a given seller
    '''
    buyers = []
    buyers_list = list()
    # get seller objects of the current active user which is required to query the Pending Transactions table

    sellers = Seller.objects.filter(seller_id_id = usn)

    #keep appending the list of buyers for every seller object

    for seller in sellers :
        buyers.extend(Pending_transactions.objects.filter(seller = seller))

    #for every buyer retrieve all his info from the Signup table

    for buyer in buyers:

        buyer_signup_obj = Signup.objects.get(usn = buyer.buyer_id_id)

        temp_dict = {
        'name'  : buyer_signup_obj.name,
        'email': buyer_signup_obj.email,
        'phone': buyer_signup_obj.phone,
        'book'  : buyer.book_name,
        'dept'  : buyer_signup_obj.dept,
        'sem'   : buyer_signup_obj.sem,
        }
        buyers_list.append(temp_dict)
    return buyers_list
```

Welcomepage (Authentication):



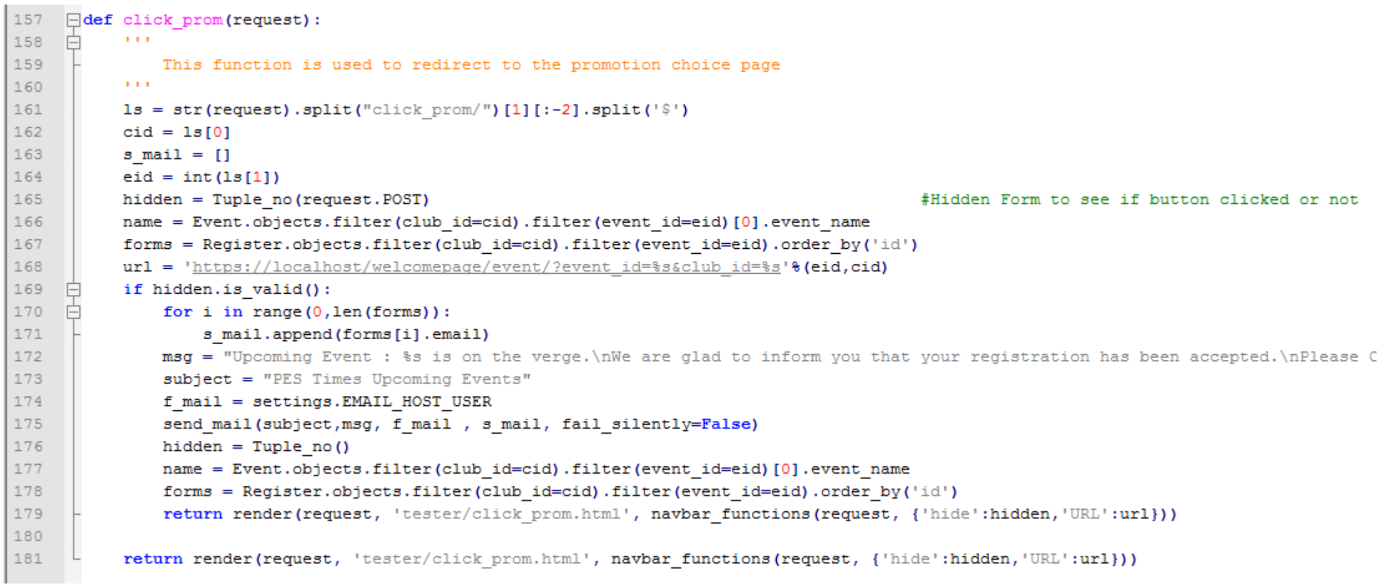
```python
        elif dict1.get("email"):
            #signup
            request.session["username"] = dict1["username"]
            request.session["usn"] = dict1["usn"]
            request.session["email"] = dict1["email"]
            request.session["mobile"] = dict1["mobile"]
            request.session["dept"] = dict1["dept"]
            request.session["sem"] = dict1["sem"]
            request.session["password1"] = dict1["password1"]
            request.session["dob"] = dict1["dob"]

            try:
                user = User.objects.get(username = usn)
            except:
                key = get_random_string(length = 8)
                request.session["key"] = key
                msg = "This is your key: " + key
                subject = "PES Times key sent"
                f_mail = settings.EMAIL_HOST_USER
                s_mail = [dict1["email"]]
                send_mail(subject, msg , f_mail , s_mail, fail_silently=False)
                extradict.update({"KEY_SENT":True,"clubs":clubs})
                return extradict
            else:
                #user exists
                extradict.update({"INTEGRITY_ERROR":True,"clubs":clubs})
                return extradict

        elif dict1.get("key"):
            if dict1.get("key") == request.session["key"]:
                #two entries are made:  One in Django User table for authentication purpose
                #                       Other in Signup table of our DB for recording user data
                new_user = User.objects.create_user(username=request.session["usn"],password=request.session["password1"])
```

Event (One tap register)



```python
def register_one_tap(request):

    #user is logged in for sure, no need to check for exceptions
    usn = request.session["usn"]
    user = Signup.objects.get(usn=usn)
    if request.method == "POST":
        parameters = request.POST
        eventid = parameters.get('eventid')
        clubid = parameters.get("clubid")
        print("************",eventid,clubid)
        required_fields = Event.objects.get(event_id=eventid,club_id=clubid).requirements
        required_fields = eval(required_fields)
        print("**********",required_fields)
        default_params = {"usn":usn,"name":"","email":"","phone":None,"dept":"","sem":""}
        for required in required_fields:
            if required == "name":
                default_params["name"] = str(user.name)
            elif required == "email":
                default_params["email"] = str(user.email)
            elif required == "phone":
                default_params["phone"] = int(user.phone)
            elif required == "dept":
                default_params["dept"] = str(user.dept)
            elif required == "sem":
                default_params["sem"] = str(user.sem)
            else:
                pass

        print("******",default_params)
        try:
            Register.objects.create(club_id = clubid,event_id = eventid,usn = usn,name = default_params["name"],email = default_params['
            #new_registration.save()
            return HttpResponse("Congrats, you have been registered!")
```

Club Admin (Event Promotion)



```python
def click_prom(request):
    '''
        This function is used to redirect to the promotion choice page
    '''
    ls = str(request).split("click_prom/")[1][:-2].split('$')
    cid = ls[0]
    s_mail = []
    eid = int(ls[1])
    hidden = Tuple_no(request.POST)                                         #Hidden Form to see if button clicked or not
    name = Event.objects.filter(club_id=cid).filter(event_id=eid)[0].event_name
    forms = Register.objects.filter(club_id=cid).filter(event_id=eid).order_by('id')
    url = 'https://localhost/welcomepage/event/?event_id=%s&club_id=%s'%(eid,cid)
    if hidden.is_valid():
        for i in range(0,len(forms)):
            s_mail.append(forms[i].email)
        msg = "Upcoming Event : %s is on the verge.\nWe are glad to inform you that your registration has been accepted.\nPlease C
        subject = "PES Times Upcoming Events"
        f_mail = settings.EMAIL_HOST_USER
        send_mail(subject,msg, f_mail , s_mail, fail_silently=False)
        hidden = Tuple_no()
        name = Event.objects.filter(club_id=cid).filter(event_id=eid)[0].event_name
        forms = Register.objects.filter(club_id=cid).filter(event_id=eid).order_by('id')
        return render(request, 'tester/click_prom.html', navbar_functions(request, {'hide':hidden,'URL':url}))

    return render(request, 'tester/click_prom.html', navbar_functions(request, {'hide':hidden,'URL':url}))
```

Software Testing

1. **Unit Testing**

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually.
Unit testing is performed by using the White Box Testing method.
Once all of the units in a program have been found to be working in the most efficient and error-free manner possible, larger components of the program can be evaluated by means of integration testing.
Unit testing was done during development phase for each Django App.

1. Event Creation Testing:
   Input: Legit data for event creation like event id, club id, etc.
   Expected Output: New entry in database and changes occurred in News Feed and Club page.
   Output matched the expectation
2. Uploading a book in Book Exchange App
   Input: Book Name and Subject
   Expected Output: New Entry in Database and book shown in up for sale books
   Output matched the expectation
3. Sign Up Feature
   Input: Legit Data given for signing up a user
   Expected Output: New Entry in the Sign up table and Django User table. User can login henceforth.
   Output matched the expectation

The test cases were created during the development stage of each of the apps separately, corresponding to the functional requirements.
Bugs that were found were all fixed such as:

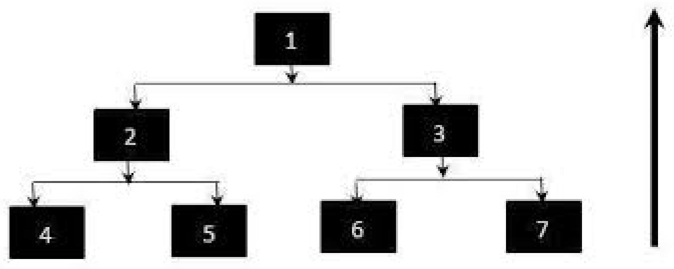
- One or two books can't be borrowed for no apparent reason: Fixed
- Adding the comments in one event could be seen in other events as well: Fixed

2. **Integration Testing**

The main function of Integration testing is to test the interfaces between the units/modules.

For our project the Integration testing means that we have to test the working of different applications (which together form one project) when combined together.

We followed Bottom up Testing for our project. In this each component at lower hierarchy is tested individually and then the components that rely upon these components are tested.



The individual modules are first tested in isolation which comprises our Unit testing. In our project we have 4 main applications named:

1. Welcome Page
2. Club
3. Club Admin
4. Book Exchange

After each application has been individually tested, mainly the testing of *views.py* and *.html* files in templates directory, we went ahead with the integration of other applications.

One thing to note is that while creating individual applications we have to assume and create some dummy data, which later on will be provided by the other applications. For ex. – While creating Club Admin we needed the SignUp information for One Click Registration. To check for that functionality we created our own SignUp table and moved ahead.

The real task was when both the Welcome Page (which has the SignUp feature) and Club Admin applications were developed, they had to be integrated. Now the dummy data were to be replaced by the actual and the one provided by the different and the authentic application.

1. Session passing across apps:
   Input: Login user in any given page
   Expected Output: User must remain logged in in all other pages of all apps
   Output matched the expectation
2. Newsfeed tabs that link apps together
   Input: Click on any given tab in the newsfeed
   Expected Output: Corresponding page in the app must be displayed
   Output matched the expectation
3. Centralized database in tester app
   Input: Make a change in the DB from any app
   Expected Output: Models must be used from tester app only and change must be reflected in a single place
   Output matched the expectation

3. **Black-Box Testing**
   Once the entire app was developed, a run through was done using the front UI only of the entire website. We names this Black-Box Testing.

As part of Black-Box testing we ignored all the back end jobs and only testing the functionalities appearing for a naïve user.

1. Authentication:
   Input: Sign a user up
   Expected Output: A key must be sent to the email id provided if it is a new user. Otherwise, a message must be displayed that this user already exists.
   Output matched the expectation
2. Creation of events using form
   Input: A club admin creates a new event using the form provided
   Expected Output: The same event is now visible to the admin as well as any user who sees the website. Club admin now has the privilege of promoting this event.
   Output matched the expectation
3. Event promotion
   Input: Club admin selects the event he wants to promote and clicks promote via email and also selects the Facebook and Twitter promotions
   Expected Output: Push mails are sent to all the users who registered. Facebook post is generated for Facebook promotion and a Twitter promotion is selected (With a hashtag) for Twitter promotion.
   Output matched the expectation