Bilkent University

Department of Computer Engineering

Perfent

Project ID: T2308

CS 491 - Senior Design Project I

Project Specification Report

Beste Güney	21901631	beste.guney@ug.bilkent.edu.tr
Bora Çün	21802838	bora.cun@ug.bilkent.edu.tr
Cemal Faruk Güney	21903474	faruk.guney@ug.bilkent.edu.tr
Çağrı Eren	21801831	cagri.eren@ug.bilkent.edu.tr
Gamze Elif Çenesiz	21902461	elif.cenesiz@ug.bilkent.edu.tr

Project Supervisor: Cevdet Aykanat

Innovation Expert: Burak Özen

Course Instructors: Tağmaç Topal, Erhan Dolak

Project Webpage: https://perfent.github.io/perfent-docs/

Table of Contents

1. Introduction

Hanging out and going to activities with one's friends, family, or any group can be an excellent way to spend time and create memories. People generally want to attend events together with others whom they wish to spend time with. These events should be interesting to them and that are a reasonable fit for their schedule. For example, if one suggests an event to their friend group that their friends are passionate about and that covers a reasonable free time slot on their friend’s schedule, most people would be happy to join such an event with their friends.

When it comes to organizing, several problems emerge such as none of the group members taking the initiative to organize, finding a proper time available for everyone, the tiresome activity of researching and choosing an event among countless websites, and finding an event that the group members will be interested in. These problems may cause them to waste a huge amount of time scheduling and browsing events on the internet to find an event that will be the best fit for the group. In addition, people can also miss an event that they would otherwise prefer to go to since they were not aware of that event. After all, it is a single person cannot know every event that exists. Perfent aims to bring solutions to these problems and transform the organization process into a more autonomous and personalized experience for groups.

1.1. Description

Perfent targets any group that wants to hang out and attend events together. It optimizes the process of finding a slot that is available for every member and also comes with event suggestions that might interest the group members. The event suggestions will combine the events from different web pages and different event providers which will also cause an improvement in the experience of browsing events. This way, the users do not have to visit several pages to find an event that fits their preferences. Perfent aims to bring incremental innovation to the process of organizing group events by optimizing the process of scheduling, organizing, and finding events that are a great fit for a group. It is aimed to enhance product performance by distinguishing features and functionality. Digital business optimization will be applied to create a better user experience and improve the productivity of the system.

The users will have their individual schedules on the system where they can indicate their busy and free time slots. For groups, these schedules will be combined to find a free slot for the whole group. In addition, based on the preferences of the group members, Perfent will recommend a proper event for the group. All group members will be notified about the free slots, the recommended event and specify whether they want to attend that event or not.

There are other applications that are based on events and some of the most known are Meetup [1] and Eventbrite [2]. Meetup is about communities hosting events in themselves and other users joining these communities and joining the events they host. Eventbrite is about users browsing local events, creating events, and buying tickets for events. There are also other applications such as AllEvents[3], Unation [4], TickPick [5], Gametime [6], TicketMaster [7], and StubHub [8]. The fundamental difference between Perfent and these applications is that Perfent focuses on groups' event experience rather than individuals and most of its feature set is around supporting groups having the best experience at easily organizing an event and attending the event.

1.2. Constraints

During the implementation and maintenance of the application, the following constraints will be considered to reach the goals of the application.

1.2.1. Implementation Constraints

- The project is designed to be implemented as a web application.
- The server and the client sides of the application will be implemented separately so that group members can progress the different parts of the project in parallel.
- For the client side of the application, HTML5, CSS, and Javascript will be used.
- To have visually appealing UI components, open-source libraries such as MUI and Bootstrap will be used.
- The recommendation system will be implemented on the server side. Because of this, the server side is expected to be coded with Python.
- For web scraping, an open-source Python library will be used. Initially, the project will scrap only one website to avoid problems related to data unification.
- Object Oriented Programming conventions will be followed during implementation.
- For machine learning implementation, open-source Python libraries such as TensorFlow, Scikit-learn, and Numpy will be used.
- To gather the calendar information of the users, the application will use Google Calendar API.
- In order to gather data quickly at the initial state of the application, click stream data will be collected from users.
- Synthetic data will be generated for testing and demo purposes.
- A relational database system will be used for data storage.
- For version control and project management, Git and Github will be used by the group members.

- Group meetings will be held on Discord platform to keep the logs.

1.2.2. Economic Constraints

- A new domain will be bought for the application.
- May need to pay for online cloud services.
- Perfent will be free to use.
- The tools to develop the application are available for the team without additional costs.

1.2.3. Social Constraints

- The application will be available both in English and Turkish. If the demand increases, it can be extended to other languages.
- Perfent will allow users to invite their friends to the site to make socializing easier.
- The application should enable users to easily connect with their friends and form groups.

1.2.4. Sustainability Constraints

- As more users join the application, the system may need bigger services and hardware.
- To survive in the market, the application needs to be innovative in terms of what it offers. Differently from other products, the application should be dynamic by updating itself regularly (both for events and user data) and will provide a recommendation for groups rather than individuals as in many current recommendation systems.

1.2.5. Time Constraints

- Web scraping will be completed as early as possible to be able to focus on the recommendation system.
- The first prototype of the application must be ready by the end of the first semester.
- The project must be completed by the end of the second semester.

1.2.6. Ethical Constraints

- Because the application aims to have a variety of users, the web scraping process should be done from a well-known trustworthy website to avoid fetching inappropriate content.

1.2.7. Legal Constraints

- Web scraping can be used for malicious purposes to fetch private or classified information. To avoid legal issues, the data will be collected discreetly and according to the terms of service of the website.

- Fetching event data from some websites may be illegal. The permissions given by the websites must be well-researched.

1.3. Professional and Ethical Issues

For the initial data collection process, users will be asked to fill out an optional form. If this form is filled by the users, the collected data will be used only for application purposes and will not be shared with outside sources. The data such as calendar and click stream information collected by users should also only be used inside of the application.

Sensitive user information such as user passwords and locations must be stored securely. To achieve this, the data will be encrypted or hashed before they are uploaded to the database.

The Perfent team wants to work in a positive working environment. To achieve this, the developers will respect each other, the work will be shared as evenly as possible, and the developers will be transparent with their progress during the weekly progress meetings.

2. Requirements

This section briefly describes the functional and non-functional requirements of the project to paint an idea of what the project will be like. The requirements mentioned here will be more detailed in the Analysis and Requirements Report.

2.1. Functional Requirements

2.1.1. Group Functionalities

- The users can create groups.
- The users can join already existing groups.
- The users can switch between their “group views” so that they can see what events are suggested to their currently chosen group.
- The groups can organize internal group activities.
- Groups can add important notes to the events they plan to attend.
- After a group has agreed to attend an event on the system, they can create a list of required items that will be brought to the event.
- Groups can assign group members to the items indicating who should bring which item.
- Item bringers will get notifications from the system so that they don't forget the items.

2.1.2. Schedule Functionalities

- The user can have a schedule view that shows events in the free slots.
- The users can import an already existing schedule from third-party applications.
- The users can combine their imported schedules if there are multiple.
- The users can synchronize their schedules if the third-party schedule is updated.
- Users can indicate between which dates they will not be available to join events so as not to disrupt the recommendation algorithm. Between the specified dates, the user will be “invisible” to the system.

2.1.3. Event Functionalities

- When a new user joins the system, the system will do a optional questionnaire (normal questions, ask previously joined events, present them with some events, and ask which ones they would attend) to gather user preferences.
- The users can browse all the upcoming and past events.
- The users can browse the events suggested to their currently chosen group.
- The users can browse the events suggested to themselves only (independent from any groups).
- The users can add constraints to their suggestion algorithm such as price, age, time, number of minimum available members, etc.
- The users can add events, artists, and venues to their “Wishlist” to receive notifications and improve suggestion accuracy.
- The users can mark events as attended.
- The users can rate events and post comments about them to improve suggestion accuracy as well as to help other users decide if the event is a recurring event.
- The system will notify the events that are the best fit for the group periodically.
- The users can propose events to their group.
- The user can view the proposed events both on the time slots on the calendar and in a different section (e.g. notifications).
- The users can optionally agree or disagree to events proposed by either group members or the suggestion algorithm.
- The user can vote among events if the members wish to choose one of many agreed events.

2.1.4. User Matching Functionalities

- If the user opts in, the system will include the user in a user-matching algorithm. With this functionality, if the user wants to meet other people to go to an event with, the system will help them find those people.

- After attending an event users can evaluate the system-recommended users they have gone to an event with as system feedback, the evaluation will not be shown to any users.
- Users can report the system-recommended users that they have gone to events with.
- Users can view the past users they have attended events with.
- Users can block users from getting recommended to them.

2.1.5. Low-priority Functionalities

- The users can create schedules directly within Perfent.
- The users can anonymously post photos and videos from the events they go to so that the viewers can have an idea of what the events are like (They might be curious about what they missed or the event might be a recurring event).
- Groups can have an event feed where they can post pictures from the event. Later these pictures can be shown in the group feed as well.
- The “admin users” can verify event runner accounts and check user reports.
- Event runners can create new events by providing all the necessary event information such as description, price, venue information, etc.
- Event runners can cancel the events they created by notifying the ticket holders and refunding the money.
- Event Runners must apply to site admins to get verified.
- The users can buy tickets for the event they are attending as a group/solo (Only applicable to events created in Perfent).
- After a group has agreed to attend an event, group members can get recommendations for the least time-consuming ways to reach the event location.

2.2. Non-Functional Requirements

2.2.1. Maintainability

The application will have the necessary documentation and tools set up to enhance maintainability which is the ease of modifying a component or a system to correct faults and improve performance or other attributes [9]. To satisfy such needs our application will use the following metrics and target a maximum of 5% code duplication threshold, a minimum of 80% unit test coverage, and a maximum cyclomatic complexity of 20 for each unit [10].

2.2.2. Availability

The application will be available for most of the time of its lifetime. Our application will aim for a minimum availability of 99% during its lifetime. Most services on the internet fall between 99% and 100% of availability and our application targets to be like one of those services at the bare minimum [11].

2.2.3. Usability

The user interface of the application should be easy to manage, simple to use, and usable. It will ensure that all of the pages of the user interface can be understood at a reasonable level and traversed in a maximum of 1 minute.

2.2.4. Safety

Any private personal information entered into the system by the user such as interests or addresses will not be disclosed to the public and will be safeguarded by the servers.

Passwords entered into the system will be hashed with effective hashing algorithms that further protect them [12].

The application will have the necessary features to ensure that users are going to hazard-free events with hazard-free users.

2.2.5. Scalability

Our servers should be able to scale when it is necessary and handle the requests incoming from 5,000 concurrent users seamlessly and without any repercussions to the users using the website and the availability of any of Perfent's functionality. It should be able to load balance the coming traffic when the traffic gets heavy since not managed traffic can cause lags in the system and lag can be a determinant factor in losing a customer [13].

2.2.6. Performance

The application will satisfy the user's waiting time expectations and prevent users from bouncing off our website. The application will target a 2-second loading time threshold with a 6-7% bounce rate for the initial (entry) loading of the website [14]. Then, for each loading of the other pages, it will target the 1-second loading time with a 6-7% bounce rate [14]. Finally, for other actions of the user in the user interface that do not include server interactions, it will target the maximum action time of 100ms.

2.2.7. Portability

The website will also be portable when viewed from devices that are not computers such as mobile devices. All of the features that operate when the website opens from a computer will also operate and will be easy to use when it is opened from a device that is not a computer. This is important because as of August 2022 53.74% of all internet traffic is coming from mobile devices instead of computers [15].

3. Risks

The development of the project brings some risks with it. The risk with the highest probability of occurring is the possibility of a change in the requirements. Most of these requirements specified in this report are the initial shape of the project after meeting with the supervisor, course advisors, and the innovation expert. As time goes on, the analysis and the requirements report will be written to detail each requirement. In the process, some initial requirements may be edited, removed, or the new ones might be added if the members see fit. Additionally, if some requirements prove to be more difficult or costly than anticipated, or simply impossible in the implementation stage; these requirements could be removed. At the time of this report, we expect to achieve our expected goals described in the next section; however, if the conditions change, there is this risk of requirement change as is the case in all software engineering projects.

Another possible risk is that the group members may underestimate the iteration one and project completion deadlines. To avoid this, we plan to set strict internal deadlines for ourselves as well as start our tasks early. This way, possible hindrances appear before the actual deadlines and we get “extra time” to make up for the time lost by these.

Unknown technical challenges in the implementation stage are also risks. Some group members have very little experience with web scraping and recommendation systems. Even after learning to implement these, there is the possibility of not being able to integrate these into the project as expected. The impact of this risk happening could be a slight change in requirements or not being able to fully complete the project by the deadline. We plan to overcome this risk by researching web scraping and recommendation systems in detail before the analysis and requirements report, and also by consulting our innovation expert who is very experienced in machine learning and recommendation systems.

4. Expected Goals

For the development of the project during the first semester we have planned these goals:

1. We plan to finish the requirement analysis and feature research by the end of October. Although, if one of the group members thinks of an exceptional idea that can contribute to the project we might try to include that at a later time than the end of October.
2. We plan to finish setting up our development environment and development tools such as Github, Jenkins, development server, etc. until mid-November.

3. We plan to do research about and at the most basic level finish the core features of the project which are the event recommendation system and event mining from the internet until mid-December.
4. Besides the core features, we also plan to implement other basic features of Perfent’s website and web server with their automated tests until mid-December.
5. For the first semester demo, we plan to have an application with basic features like creating and joining groups or viewing events, etc., and the most basic state of the core features such as event recommendations, collecting user data, event mining from the internet, finding available group time optimization algorithms.
6. We also plan to do manual tests before the demo to ensure that the basic and core features of the project are working correctly.

For the second semester, we have also planned our goals however, it is hard to make deductions about a time that is so distant.

7. During February, March, April, and May we plan to continue developing and doing research about the basic features and core features of the project.
8. We plan to start the features that have low priority at the end of March. Although this might change according to our progress on the basic features and core features of the project. If necessary low priority features will be removed from development.
9. We plan to switch to synthetically generated data until the end of April. From that point, our development and testing will be done using synthetic data which will allow us to see how our project performs under synthetic data.
10. At the end of the year, we will have a project that has its core, basic, and low-priority features finished and working with generated synthetic data.

All of these expected goals are subject to change and may change according to the progression of the project, this is just a rough plan.

5. References

1) “We are what we do,” *Meetup*. [Online]. Available: https://www.meetup.com/. [Accessed: 16-Oct-2022].
2) *Eventbrite*. [Online]. Available: https://www.eventbrite.com/. [Accessed: 16-Oct-2022].
3) “Discover events happening in your city,” *AllEvents.in*. [Online]. Available: https://allevents.in/. [Accessed: 16-Oct-2022].
4) *UNATION*. [Online]. Available: https://www.unation.com/. [Accessed: 16-Oct-2022].
5) “No-fee sports, concert, theater tickets | tickpick.” [Online]. Available: https://www.tickpick.com/. [Accessed: 16-Oct-2022].
6) “Last-minute sports, Music & Shows Tickets,” *Gametime*. [Online]. Available: https://gametime.co/. [Accessed: 16-Oct-2022].
7) “Ticketmaster: Buy verified tickets for concerts, sports, theater and events.” [Online]. Available: https://www.ticketmaster.com/. [Accessed: 16-Oct-2022].
8) “Buy sports, concert and theater tickets on StubHub!,” *stubhub.com*. [Online]. Available: https://www.stubhub.com/. [Accessed: 16-Oct-2022].
9) C. Chen, R. Alfayez, K. Srisopha, B. Boehm, and L. Shi, “Why is it important to measure maintainability and what are the best ways to do it?,” *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, 2017.
10) I. Heitlager, T. Kuipers and J. Visser, "A Practical Model for Measuring Maintainability," 6th International Conference on the Quality of Information and Communications Technology (QUATIC 2007), 2007, pp. 30-39, doi: 10.1109/QUATIC.2007.8.
11) B. Thorne, “Four nines and Beyond: A guide to high availability infrastructure,” *Work Life by Atlassian*, 26-Oct-2020. [Online]. Available: https://www.atlassian.com/blog/statuspage/high-availability. [Accessed: 16-Oct-2022].
12) “Where do websites store passwords?,” *The JavaScript Diaries*, 18-Nov-2019. [Online]. Available: https://www.jsdiaries.com/where-do-websites-store-passwords/. [Accessed: 17-Oct-2022].
13) C. Song, “Scalable systems 101,” *Educative*. [Online]. Available: https://www.educative.io/blog/scalable-systems-101. [Accessed: 17-Oct-2022].
14) “Does page load time really affect bounce rate? - pingdom,” *pingdom.com*. [Online]. Available: https://www.pingdom.com/blog/page-load-time-really-affect-bounce-rate/. [Accessed: 17-Oct-2022].
15) J. Gaubys, “What percentage of internet traffic is mobile? [Sep '22 UPD],” *Oberlo*. [Online]. Available: https://www.oberlo.com/statistics/mobile-internet-traffic#:~:text=As%20of%20August%202022%2C%2053.74,46.26%20percent%20coming%20from%20desktops. [Accessed: 17-Oct-2022].
.