
Project Report Appendices

Tamas Fekete (266931 ICT)
Dorcia Fiona Dinesh (280145 ICT)
Zuzanna Maria Czepukojc (280163 ICT)
Lau Ravn Nielsen (280337 ICT)

Supervisor:

Jakob Knop Rasmussen (JKNR)
Jan Munch Pedersen (JPE)

Software Engineering

3rd semester

Table of Contents

| | |
|--|-----------|
| Appendix A: Project description..... | 4 |
| <i>Table of contents</i> | 5 |
| <i>List of tables and figures</i> | 5 |
| 1. Background description..... | 6 |
| 2. Problem statement..... | 7 |
| 2.1 Main problem | 7 |
| 2.2 Sub problems..... | 7 |
| 3. Definition of purpose..... | 7 |
| 4. Methodology..... | 8 |
| Project development | 8 |
| Software development | 9 |
| 5. Time schedule..... | 10 |
| 6. Risk assessment..... | 10 |
| 7. Sources of information | 11 |
| Appendix B: Activity diagrams..... | 13 |
| Create account..... | 13 |
| Create event | 14 |
| List of requests | 15 |
| List of events | 15 |
| Update event information | 16 |
| Cancel event | 17 |
| Appendix C: Use case descriptions | 18 |
| Use case description – Create Event..... | 18 |
| Use case description – Create account. | 19 |
| Use case description – Manage Event. | 21 |
| Use case description - View history..... | 22 |
| Use case description – Manage account..... | 23 |
| Appendix D: Sequence diagrams | 24 |

| | |
|---|-----------|
| Create event sequence diagram..... | 24 |
| System sequence diagram for Create event | 24 |
| Appendix E: Class diagrams..... | 25 |
| Second tier | 25 |
| Third tier | 27 |
| Appendix F: Java sockets implementation | 28 |
| Appendix G: C# listener | 29 |
| Appendix H: System tests | 30 |
| Test cases – Create account | 30 |
| Test cases – Create event..... | 31 |
| Test cases – Search event..... | 32 |
| Test cases – View account details. | 33 |
| Test cases – View event details..... | 34 |
| Test cases – View history – Branch 1 – List of events | 34 |
| Test Cases – View history – branch 2 – List of requests. (Not implemented)..... | 35 |
| Appendix I: User manual | 36 |

Appendix A: Project description

SEP3 Project description

Dining with strangers

Members:

Lau Ravn Nielsen (280337)
Tamas Fekete (266931)
Dorcia Fiona Dinesh (280145)
Zuzanna Maria Czepukojc (280163)

Supervisors:

Jakob Knop Rasmussen
Jan Otto Munch Pedersen

Characters:

12.435
14.464 (with spaces)

Deadline:

07 / 10 / 2019
12:00

Table of contents

| | |
|----------------------------------|-----------|
| 1. Background description | 6 |
| 2. Problem statement | 7 |
| 2.1 Main problem | 7 |
| 2.2 Sub problems | 7 |
| 3. Definition of purpose | 7 |
| 4. Methodology | 8 |
| Project development | 8 |
| Software development | 9 |
| 5. Time schedule | 10 |
| 6. Risk assessment | 10 |
| 7. Sources of information | 11 |

List of tables and figures

| | |
|---------------------------|----|
| TABLE 1 - RISK ASSESSMENT | 10 |
| FIGURE 1 - MILESTONE-LINE | 10 |

1. Background description

Humans have evolved into social beings out of the necessity of being dependent with each other in order to survive harsh circumstances. In today's world as well, people continue to have a need to associate with others. Finding new ways to develop strong social connections is integral to the wellbeing of humans. The lack of such connections can lead to many issues, including loneliness. Loneliness in its essence is a negative feeling one may experience when our social needs aren't met by our current social circle (Blackdoginstitute.org.au, 2018). The causes for loneliness fluctuate over the duration of one's life. This is due to the different needs at different ages (Department for Digital, Culture, Media and Sport, 2019).

However, a few causes could be one's personality, circumstances, environmental conditions, local social networks, social media, the way we live and work, or life events (Department for Digital, Culture, Media and Sport, 2019).

Loneliness is often described as an epidemic (Hrsa.gov, 2019). It's described as an epidemic due to the large share of the population which is affected by it. In a 2018 survey from Australia 50 percent of the respondents reported feeling lonely (Dingle, 2019). In Denmark, 12% of young people in the age group 16-24 years feel very lonely on the UCLA Loneliness Scale, with the majority being women (15%) and men (10%) (Lasgaard, Christansen and Friis, 2019). A recent study of adults of all ages in San Diego, California concluded that 76% of the respondents felt moderately to high levels of loneliness on the UCLA-3 scale (Ardelt and Ferrari, 2018).

Whenever one begins to feel lonely, he or she will experience an increased feeling of vulnerability, which in turn will take a toll on both the body and mind (Blackdoginstitute.org.au, 2018). There's significant evidence supporting the link between various mental problems such as high-stress levels, anxiety, poor sleep quality, self-harm, depression, and extended periods of psychological suffering (Lasgaard, Christansen and Friis, 2019) (T. Cacioppo and Elizabeth Hughes, 2006).

Loneliness increases the risk for dementia, likely through these mechanisms, however, the absence of social interaction itself may also be a primary factor in that social stimulation can help maintain brain health (Cacioppo & Hawkley, 2009; Cacioppo et al., 2014)

However, it's not clear whether or not loneliness is the cause or the effect of the aforementioned mental problems. Nevertheless, it's clear that loneliness goes hand-in-hand with various mental problems (Lasgaard, Christansen and Friis, 2019).

A few ways to combat the feeling of loneliness, is introspection, connection and interaction. Meeting and interacting with strangers, or interacting with people one already knows, is the key to fighting one's loneliness (Carlton, 2019). One of the few initiatives the Danish government has deployed in 2016 in its fight against loneliness, was Denmark-eats-together (MaryFonden.dk, 2016). Before launching the campaign, Maryfonden conducted a survey in order to explore the waters beforehand. The survey showed that 23,6% of the respondents missed someone to dine with, and 44% of the respondents would dine with someone they barely know (Korsgaard, 2016). Socializing has various impacts on the human body such as brain health, mental health and it may lead to a stronger immune system. It is proven by a 2017 study from Dublin, Ireland that after the age of 50 more socialized individuals had better processing speed, memory, verbal fluency and

emotional regulation. Furthermore, it may reduce the risk of dementia and symptoms of Alzheimer's more frequently than the nonsocial counterparts.

For mental health, socializing may help to develop a sense of belonging and other supportive feelings and beliefs which is something both introverts and extroverts want to achieve. When people feel better self-esteem, it may help one reduce the likeliness of anxiety disorders and depression. It is proven that laughter itself has health benefits, hence it reduces stress and improves sleep quality.

On the other hand, negative or so-called "toxic" relationships lead people to the opposite side of the scale, it might lower one's self-esteem and might higher the likeliness of mental health disorders. Lastly, socializing can lead one to have a stronger immune system.

As it was mentioned in the mental health section, positive social life can help reduce stress. If one can release stress, then it is unlikely to have chronic stress which can affect the hormone system as well as it may weaken one's immune system.

Moreover, stress from negative or no social relationships may lead to poor/negative health habits such as alcohol abuse and passive lifestyle which may have a negative impact on the immune system as well.

On the other side, healthy social bonds can help develop positive health habits, for example sports activities, general motivation in everyday life and basic hygiene which improves body resistance.

2. Problem statement

2.1 Main problem

Loneliness is becoming a big problem in Denmark. Based on studies, Danes often feel lonely however, more Danes are susceptible to feel lonely when they have no one else to dine with.

2.2 Sub problems

- What type of services should be provided for the users to decrease the feeling of loneliness?
- What data about the user is necessary to provide the services?
- Who are the relevant recipients of the service?
- Which type of services must be available for different users?
- How can forming social bonds be encouraged?

3. Definition of purpose

The purpose of this project is to create a tool which helps people to develop social bonds through dining events, which may benefit the mental health of the participants.

4. Methodology

Project development

We have decided to manage our project with the help of scrum. Scrum will be used to ensure that the team stays on track both in terms of time but also in the terms of where the product. This will be done by running sprints spanning 2-weeks which will change to 1-week sprints during the project period.

Each sprint will start by planning the sprint itself. The planning will be conducted with the entire group present. During the planning of a 2-week sprint, the team will have to plan a total of 90 hours of work for that given sprint. Furthermore, a sprint goal will be set as an objective for the developer team to work towards.

Every day of the sprint, a scrum meeting will be held. During this meeting, the team will forecast the upcoming sprint work and inspect the work since the last daily meeting. The inspecting will be done to inspect if the development team is on track to complete the work in the sprint backlog, this will be visualized with burndown-charts.

Once a sprint is over the team will conduct a sprint review with everyone present, including the scrum master and the product owner. During the review, the team will discuss what went well during the sprint, eventual problems that they experienced and how those problems were solved. Afterwards, the team showcases what work has been “done” and will answer questions about the specific work.

Then the product owner will discuss the current state of the product backlog.

Lastly, the entire group discusses what to do next, for the sprint review to provide valuable input for the next sprint planning.

The last thing in a sprint will be the sprint retrospective. The sprint retrospective will be used to reflex upon, how the team did in the previous sprint. The team will reflex upon the people, relationships, and tools used in the previous sprint. Additionally, the team will identify what went well or perhaps not so well and come up with improvements. Lastly, the team will come up with a plan for implementing the improvements they have come up with.

The scrum master:

Dorcia Fiona Dinesh

The product owner:

Lau Ravn Nielsen

The development team:

Lau Ravn Nielsen

Tamas Fekete
Dorcia Fiona Dinesh
Zuzanna Maria Czepukojc

Software development

Unified process will be used as a software development process alongside of scrum. Unified process is an iterative and incremental development process. During the development of a piece of software with unified process, the team will go through inception, elaboration, construction, and transition phases.

During the inception phase, the team will be focused on describing the domain and the requirements for the software. Furthermore, a plan for the elaboration phrase will be made. Lastly, a plan for risks revolving the project will be made and a plan to mitigate those risks.

During the elaboration phase the team will focus on conducting thorough analysis and design on the system requirements. With the result being a finished domain model. Additionally, the team will develop the core architecture and data modelling during the elaboration.

When the team gets to construction, their focus will be on implementation and getting ready for the final hand-in. This will be conducted with the help of the foundation laid out in the elaboration phase.

During the transition phase the team will be focused on finishing the nitty gritty things that are left un-done after the previous phases.

5. Time schedule

SEP3 is a total of 10 ECTS and as a rule of thumb, each student is expected to work an average of 27.5 hours per ECTS. This totals to 1100 hours for our group.

Below is a milestone-line, which has been made in order to outline the project and the various milestones herein.

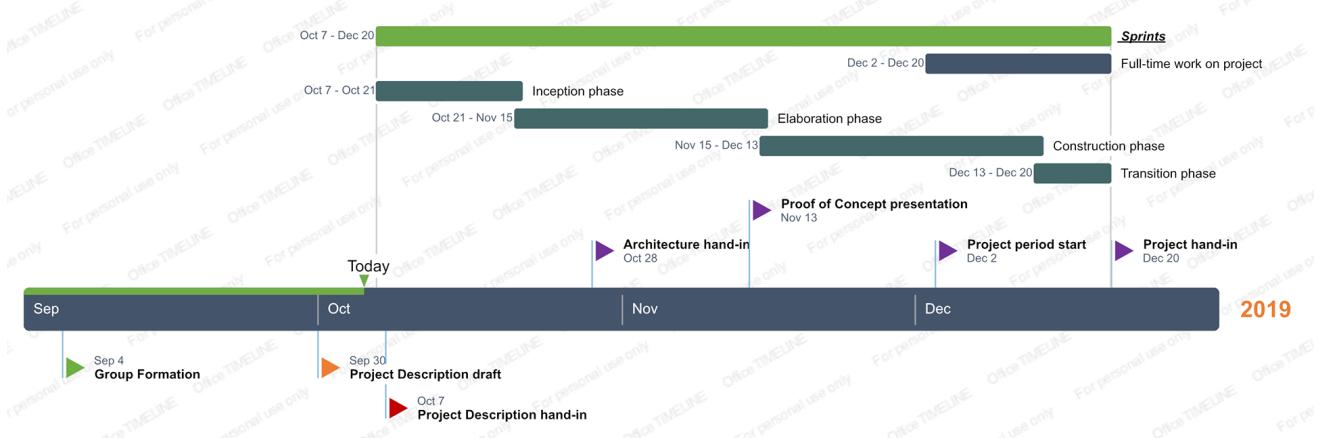


Figure 1 - Milestone-line

6. Risk assessment

| Risk | Description | Likelihood Scale | Severity Scale | Responsive Action | Identifiers | Responsible |
|------|--|------------------|----------------|---|--|-------------|
| 1 | A user decides to cancel event | 2,5 | 5 | Inform the users about the cancelation | The dining event is cancelled | Dorcia |
| 2 | No one signs into the application | 1 | 5 | Advertisement | Few users | Lau |
| 3 | Somebody signs into the system with fake information | 2 | 5 | Terminate the account | Failed to authenticate user's credentials | Zuzanna |
| 4 | Unable to recognise location | 2 | 4 | Verify location | User cannot find the location of his/her host. | Dorcia |
| 5 | The user violates the terms of service | 3 | 5 | Terminate the account | User is reported by others | Tamas |
| 6 | A user makes a fake event | 5 | 5 | Terminate the event and warn the participants | User is misled by the fake event | Zuzanna |

Table 1 - Risk assessment

7. Sources of information

Blackdoginstitute.org.au. (2018). What is loneliness and how can we overcome it? Explained. [online] Available at: <https://www.blackdoginstitute.org.au/news/news-detail/2018/11/12/what-is-loneliness-and-how-can-we-overcome-it-explained> [Accessed 27 Sep. 2019].

Department for Digital, Culture, Media and Sport (2019). A connected society - A strategy for tackling loneliness. London: Department for Digital, Culture, Media and Sport, p.17. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/750909/6.4882_DCMS_Loneliness_Strategy_web_Update.pdf [Accessed 27 Sep. 2019]

Hrsa.gov. (2019). The "Loneliness Epidemic" | Official web site of the U.S. Health Resources & Services Administration. [online] Available at: <https://www.hrsa.gov/eneews/past-issues/2019/january-17/loneliness-epidemic> [Accessed 27 Sep. 2019].

Dingle, D. (2019). Tackling the growing problem of loneliness and isolation. [online] UQ News. Available at: <https://www.uq.edu.au/news/article/2019/07/tackling-growing-problem-of-loneliness-and-isolation> [Accessed 27 Sep. 2019].

Lasgaard, M., Christansen, J. and Friis, K. (2019). Ensomhed blandt unge. [ebook] Region Midtjylland, pp.14, 15, 24, 25, 26. Available at: <https://www.maryfonden.dk/files/Ensomhed%20blandt%20unge.pdf> [Accessed 27 Sep. 2019].

Ardelt, M. and Ferrari, M. (2018). Effects of wisdom and religiosity on subjective well-being in old age and young adulthood: exploring the pathways through mastery and purpose in life. [online] Cambridge.org. Available at: <https://www.cambridge.org/core/journals/international-psychogeriatrics/article/effects-of-wisdom-and-religiosity-on-subjective-wellbeing-in-old-age-and-young-adulthood-exploring-the-pathways-through-mastery-and-purpose-in-life/C59037134A43FD9507E06AFD45580E84> [Accessed 27 Sep. 2019].

T. Cacioppo, J. and Elizabeth Hughes, M., 2006. Loneliness As A Specific Risk Factor For Depressive Symptoms: Cross-Sectional And Longitudinal Analyses. [online] Citeseerx.ist.psu.edu. Available at: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.321.3922&rep=rep1&type=pdf>> [Accessed 27 September 2019].

Maryfonden.dk. 2016. Hver 4. Dansker Savner Nogen At Spise Sammen Med | Maryfonden.Dk. [online] Available at: <<https://www.maryfonden.dk/da/danmark-spiser-sammen-pm>> [Accessed 30 September 2019].

Korsgaard, K., 2016. Danmark Spiser Sammen. [PDF] MaryFonden, p.14. Available at: <https://www.maryfonden.dk/files/files/Danmark%20spiser%20sammen_final.pdf> [Accessed 30 September 2019].

Carlton, K. (2019). What is loneliness? - UChicago Medicine. [online] Uchicagomedicine.org. Available at: <https://www.uchicagomedicine.org/forefront/health-and-wellness-articles/2019/february/what-is-loneliness> [Accessed 27 Sep. 2019].

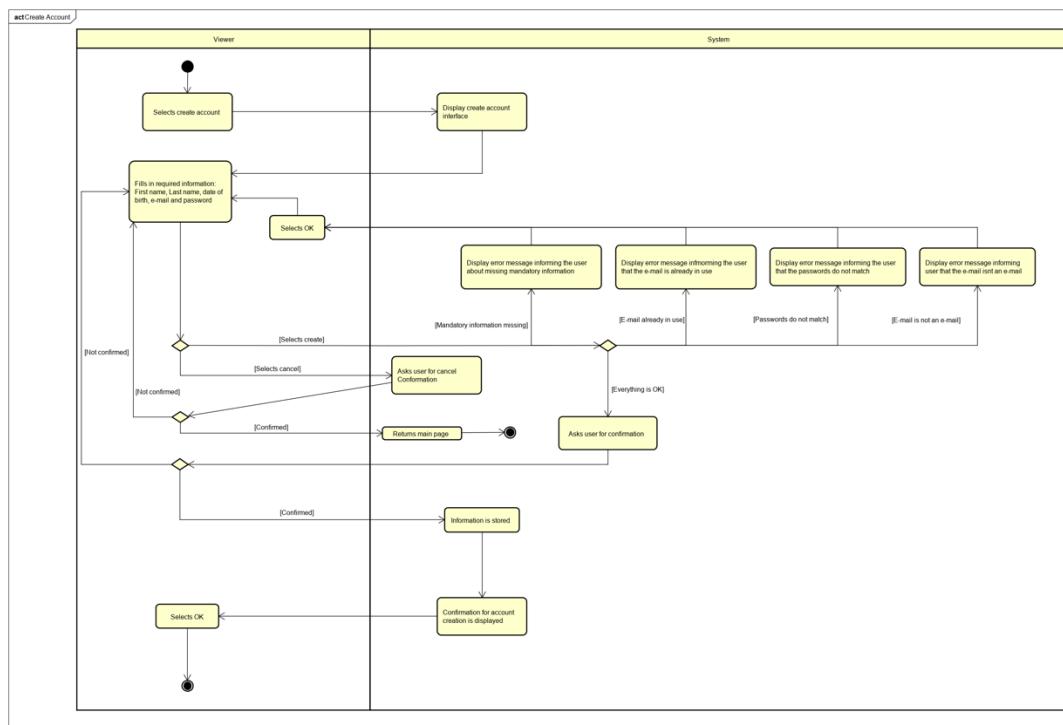
Sörman DE, e. (2019). Social relationships and risk of dementia: a population-based study. - PubMed - NCBI. [online] Ncbi.nlm.nih.gov. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/25779679> [Accessed 29 Sep. 2019].

R, M. (2010). The therapeutic value of laughter in medicine. - PubMed - NCBI. [online] Ncbi.nlm.nih.gov. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/21280463> [Accessed 29 Sep. 2019].

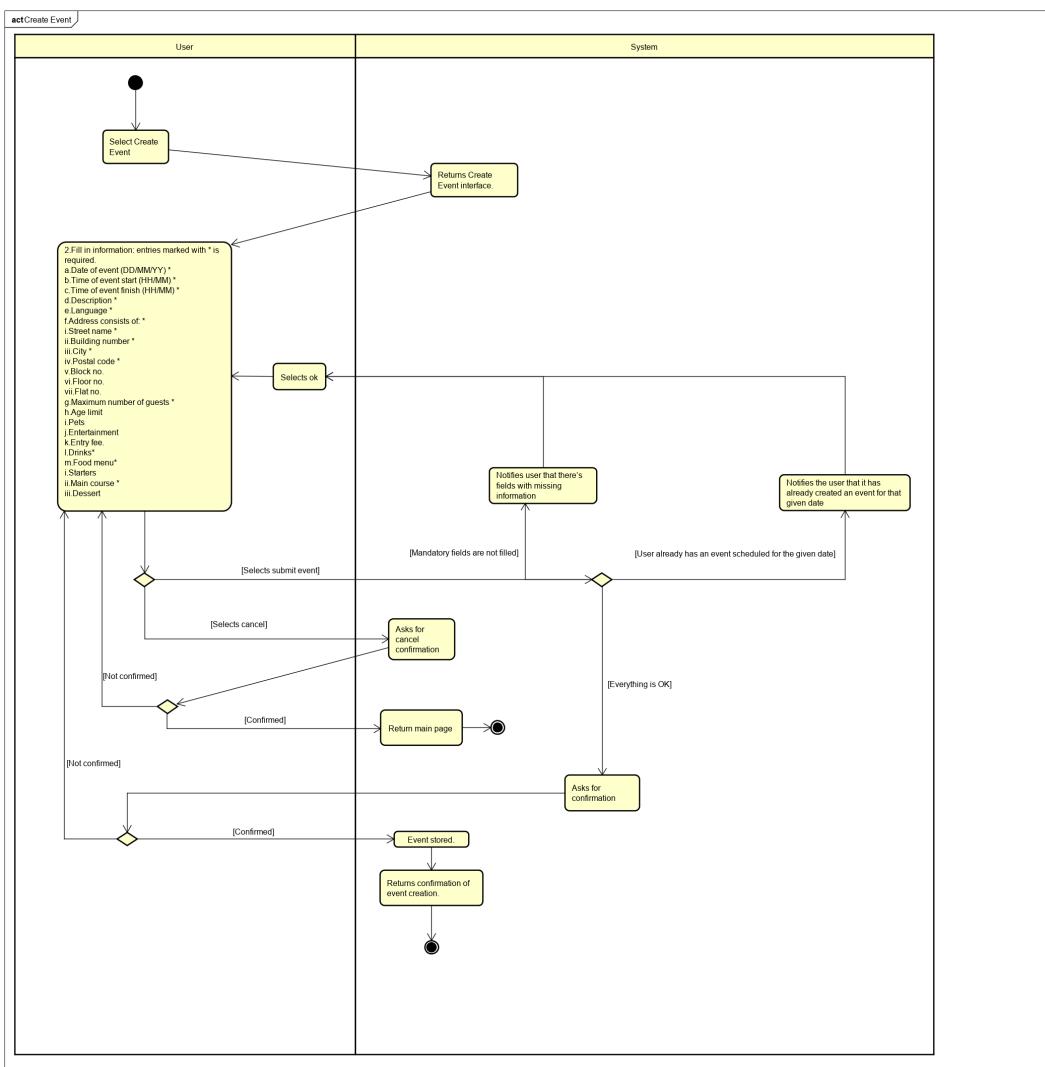
Umberson, D. and Karas Montez, J. (2019). Social Relationships and Health: A Flashpoint for Health Policy.

Appendix B: Activity diagrams

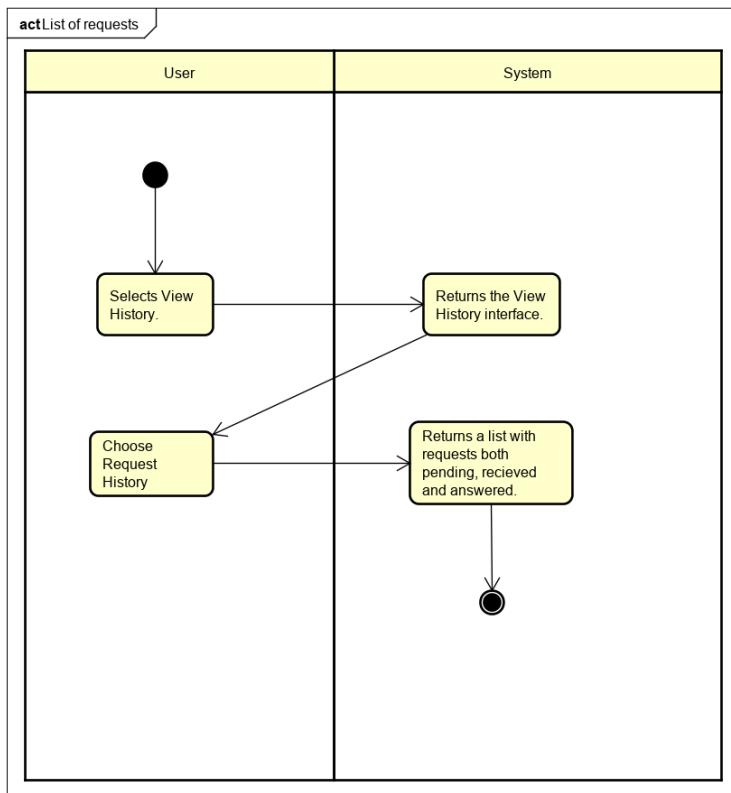
Create account



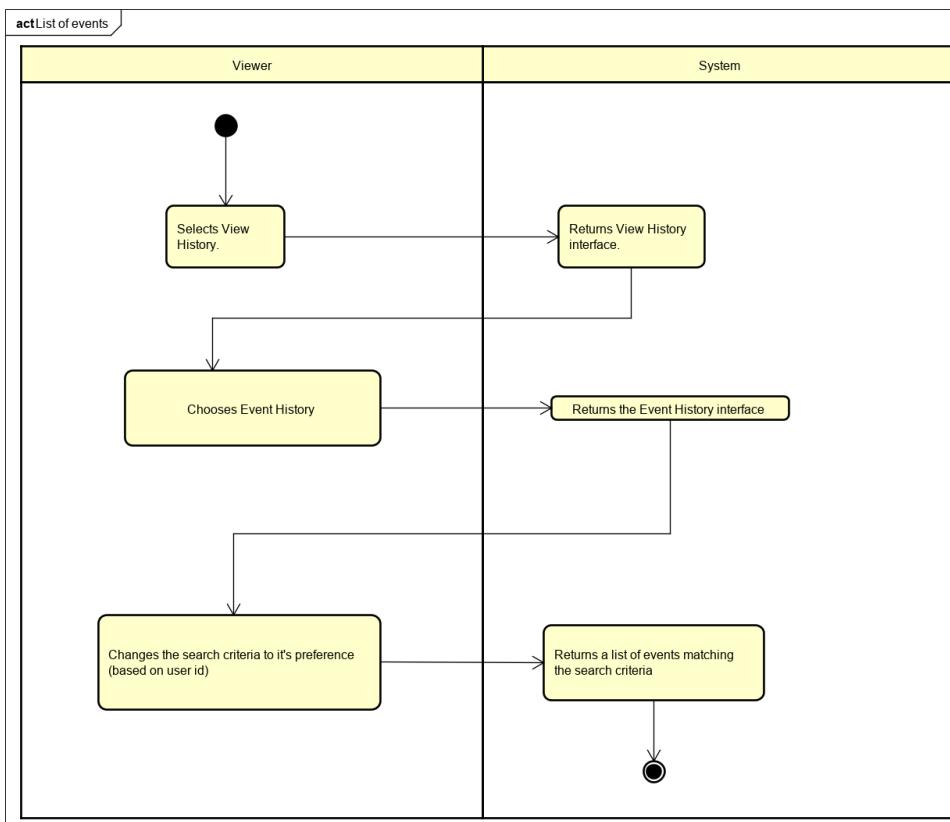
Create event



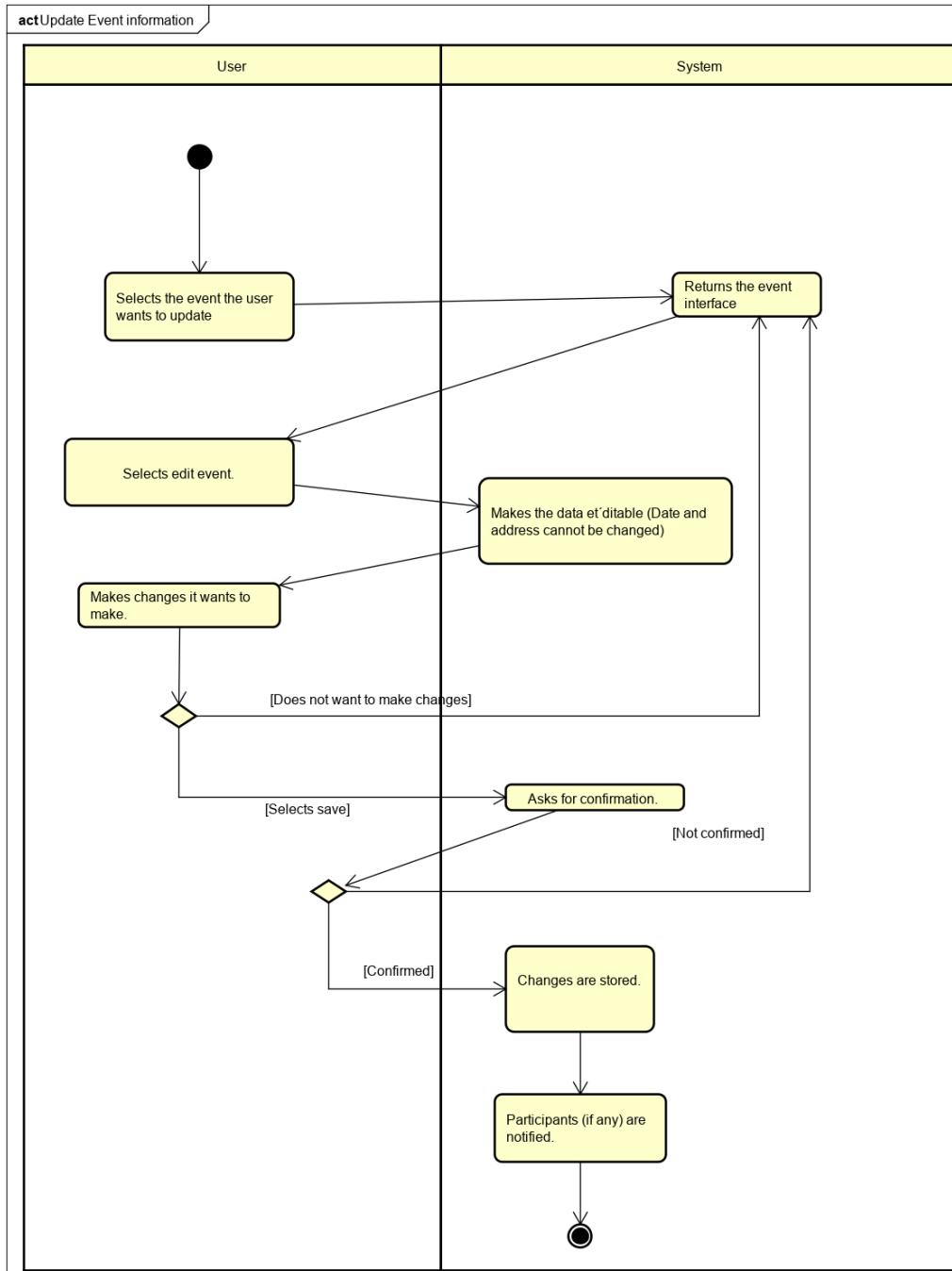
List of requests



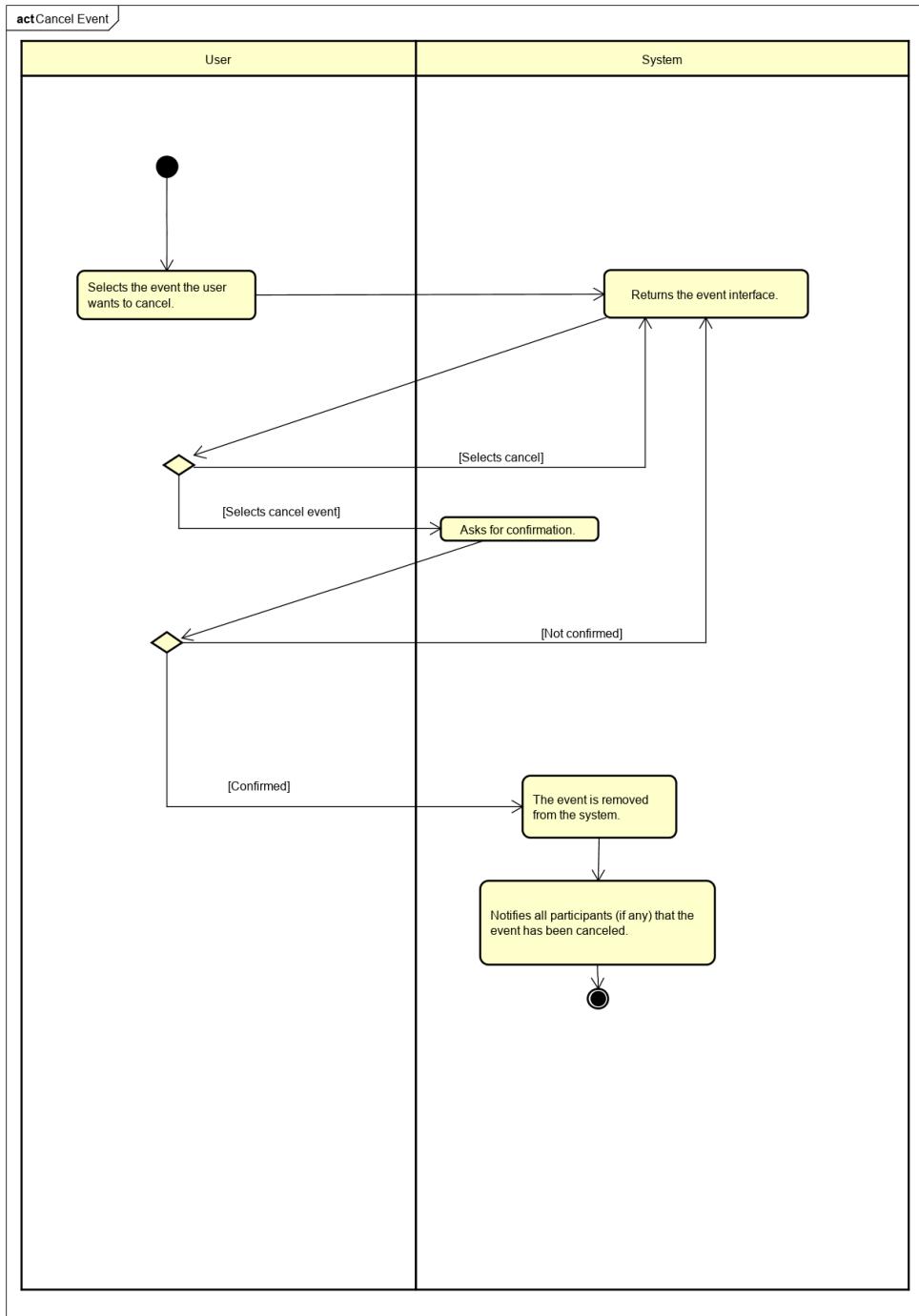
List of events



Update event information



Cancel event



Appendix C: Use case descriptions

Use case description – Create Event.

| | |
|----------------------------|--|
| Use Case Section | |
| Use Case Name | Create Event |
| Scope | Dining with strangers |
| Level | User goal |
| Primary Actor | User |
| Stakeholders and interests | User – Wants the ability to create events such that they can dine with strangers. Furthermore, other users are interested in events being created so there's event they can participate in. |
| Preconditions | User is authenticated by the system |
| Success Guarantee | The event is created and stored in the system. |
| Main Success Scenario | <ol style="list-style-type: none"> 1. User selects create event. 2. System returns create event interface. 3. Fill in information: entries marked with * is required. <ol style="list-style-type: none"> a. Date of event (DD/MM/YY) * b. Time of event start (HH/MM) * c. Time of event finish (HH/MM) * d. Description * e. Language * f. Address consists of: <ol style="list-style-type: none"> i. Street name * ii. Building number * iii. City * iv. Postal code * v. Block no. vi. Floor no. vii. Flat no. g. Maximum number of guests * h. Age limit i. Pets j. Entertainment k. Entry fee. l. Drinks* m. Food menu* <ol style="list-style-type: none"> i. Starters ii. Main course * iii. Dessert 4. User selects submit event. 5. System asks for confirmation 6. User confirms the confirms the confirmation. 7. The event is stored in the system. 8. System returns confirmation of event creation. |
| Extensions | Create event – User wants to cancel. |

| | |
|-------------------------------------|--|
| | <p>1-3 base sequence.</p> <p>4. User selects cancel.</p> <p>5. System asks for confirmation.</p> <p>6. User selects ok or cancel.</p> <p>User selects cancel.</p> <p>6.1 System returns user to the create event interface.</p> <p>User selects ok.</p> <p>6.1 System returns the user to the main page.</p> <p>Create event – Mandatory fields are not filled.</p> <p>1-4 base sequence</p> <p>5. System notifies user that there's fields with missing information.</p> <p>6. user selects ok</p> <p>7. System returns the user to the create event interface.</p> <p>Create event – User already has an event scheduled for the given date.</p> <p>1-4 base sequence.</p> <p>5. System notifies the user that it has already created an event for that given date.</p> <p>6. User selects ok.</p> <p>7. System returns the user to the create event interface.</p> <p>Create event – User declines confirmation.</p> <p>1-5 base sequence.</p> <p>6. User declines the confirmation.</p> <p>7. System returns the user to the create event interface.</p> |
| Special Requirements | |
| Technology and Data variations list | |
| Frequency of occurrence | Often – Whenever a user wants to create an event. |
| Open issues | <ul style="list-style-type: none"> • Server timeouts are not handled. |

Use case description – Create account.

| | |
|----------------------------|--|
| Use Case Section | |
| Use Case Name | Create account. |
| Scope | Dining with strangers. |
| Level | User goal. |
| Primary Actor | Viewer. |
| Stakeholders and interests | Viewer: Wants to make a new account so he/she can use the system to its fullest potential. |
| Preconditions | Application is installed and opened on the main page. |

| | |
|-----------------------|--|
| Success Guarantee | The viewer's information is stored in the system. |
| Main Success Scenario | <p>1. Viewer selects 'create account'.</p> <p>2. Create account interface is displayed.</p> <p>3. Viewer fills in the required information: first name, last name, e-mail, date of birth, and password(twice). Everything is mandatory.</p> <p>4. The Viewer selects save.</p> <p>5. System asks if the information is correct.</p> <p>6. Viewer confirms.</p> <p>7. The information is stored.</p> <p>8. Confirmation for account creation is displayed.</p> |
| Extensions | <p>Create account - At any time, Viewer wants to cancel.</p> <p>1-2. Base sequence.</p> <p>3. Viewer selects cancel.</p> <p>4. System ask for confirmation.</p> <p style="padding-left: 2em;">Viewer confirms the cancelation:</p> <p>5.1 Viewer confirms the cancelation.</p> <p>6. Viewer selects ok.</p> <p>7. Viewer is returned to the login page.</p> <p style="padding-left: 2em;">Viewer denies the confirmation:</p> <p>5.2 Viewer denies the confirmation.</p> <p>6. System returns user to the create account interface.</p> <p>Create account – Information is incorrect.</p> <p>1-5. Base sequence.</p> <p>6. Viewer denies the confirmation</p> <p>7. System returns the account creation page.</p> <p>Create account - Passwords do not match:</p> <p>1-4. Base sequence.</p> <p>5. System displays error informing the Viewer that the passwords do not match.</p> <p>6. Viewer selects ok.</p> <p>7. Viewer is returned to the create account interface.</p> <p>Create account - E-mail already in use:</p> <p>1-4. Base sequence.</p> <p>5. System displays error informing the viewer that the e-mail is already in use.</p> <p>6. Viewer selects ok.</p> <p>7. Viewer is returned to the create account interface.</p> <p>Create account – Mandatory information missing:</p> <p>1-4. Base sequence.</p> <p>5. System displays error informing the user that there's missing mandatory information.</p> <p>6. Viewer selects ok.</p> |

| | |
|-------------------------------------|---|
| | <p>7. Viewer is returned to the create account interface.</p> <p>Create account – E-mail is not an e-mail:</p> <ol style="list-style-type: none"> 1-4. Base sequence. 5. System displays error informing the user that the entered e-mail is not an e-mail. 6. Viewer selects ok. <p>Viewer is returned to the create account interface.</p> |
| Special Requirements | |
| Technology and Data variations list | |
| Frequency of occurrence | Whenever a Viewer wants to create an account. |
| Open issues | |

Use case description – Manage Event.

| | |
|----------------------------|--|
| Use Case Section | |
| Use Case Name | Manage Event |
| Scope | Dining with strangers. |
| Level | Summary |
| Primary Actor | User |
| Stakeholders and interests | User wants to manage the events they've created for them to keep them up to date. Other users participating in the events will have an interest for updated information. |
| Preconditions | The user has created an event and is viewing the Event History. |
| Success Guarantee | The user changes to events has been stored and participants notified if there are any. |
| Main Success Scenario | <p>Branch 1 – Cancel event</p> <ol style="list-style-type: none"> 1. User selects the event the user wants to cancel. 2. System returns the event interface. 3. User selects cancel event. 4. System asks for confirmation. 5. User responds to the confirmation. 6. The event is removed from the system. 7. The system notifies the participants (if any) that the event has been canceled. <p>Branch 2 – Update Event information</p> <ol style="list-style-type: none"> 1. User selects the event the user wants to update. 2. System returns the event interface. 3. User selects edit event. 4. System makes the data editable (<i>Date and address cannot be changed</i>). 5. User makes the changes it wants to make. 6. User selects save. 7. System asks for confirmation. 8. User selects ok. 9. Changes are stored in the system. 10. Participants (if any) are notified by the system. |

| | |
|-------------------------------------|--|
| Extensions | <p>Branch 1 - User doesn't want to cancel.</p> <ol style="list-style-type: none"> 1. 1-2 base sequence 2. User selects cancel <p>Branch 2 - User doesn't want to make any changes.</p> |
| Special Requirements | |
| Technology and Data variations list | |
| Frequency of occurrence | |
| Open issues | |

Use case description - View history.

| | |
|----------------------------|---|
| Use Case Section | |
| Use Case Name | View history |
| Scope | Dining with strangers. |
| Level | User goal. |
| Primary Actor | User |
| Stakeholders and interests | User wants to see a history of everything that the user has done within the community. This includes seeing a list of events, own events, pending requests and requests from other users to join events. |
| Preconditions | User is authenticated. |
| Success Guarantee | The user is returned a list of the preferred history. |
| Main Success Scenario | <p>Branch 1 – List of events.</p> <ol style="list-style-type: none"> 1. User selects View History 2. System returns the View History interface. 3. The user chooses Event History. 4. The system returns the Event History interface. 5. User changes the search criteria to its preference. (Can be searched by: When; <i>All, upcoming, and past events</i>. Furthermore, by Host; <i>Myself or everyone</i>. <i>All and everyone are the default search criteria.</i> 6. The system returns a list of events matching the search criteria. <p>Branch 2 – List of requests</p> <ol style="list-style-type: none"> 1. User selects View History. 2. System returns the View History interface. 3. User chooses Request History. 4. The system returns a list with requests both pending, received, and answered. |
| Extensions | |
| Special Requirements | |

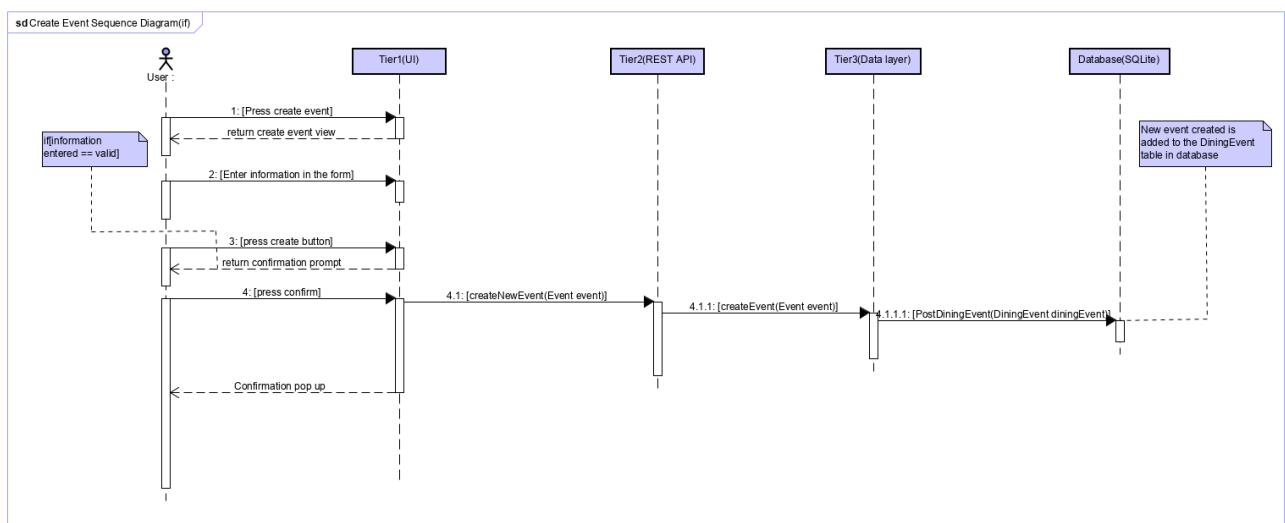
| | |
|-------------------------------------|--|
| Technology and Data variations list | |
| Frequency of occurrence | Whenever a user must view its history of requests or events. This also must be done whenever a user wants to manage events or requests respectively. |
| Open issues | |

Use case description – Manage account

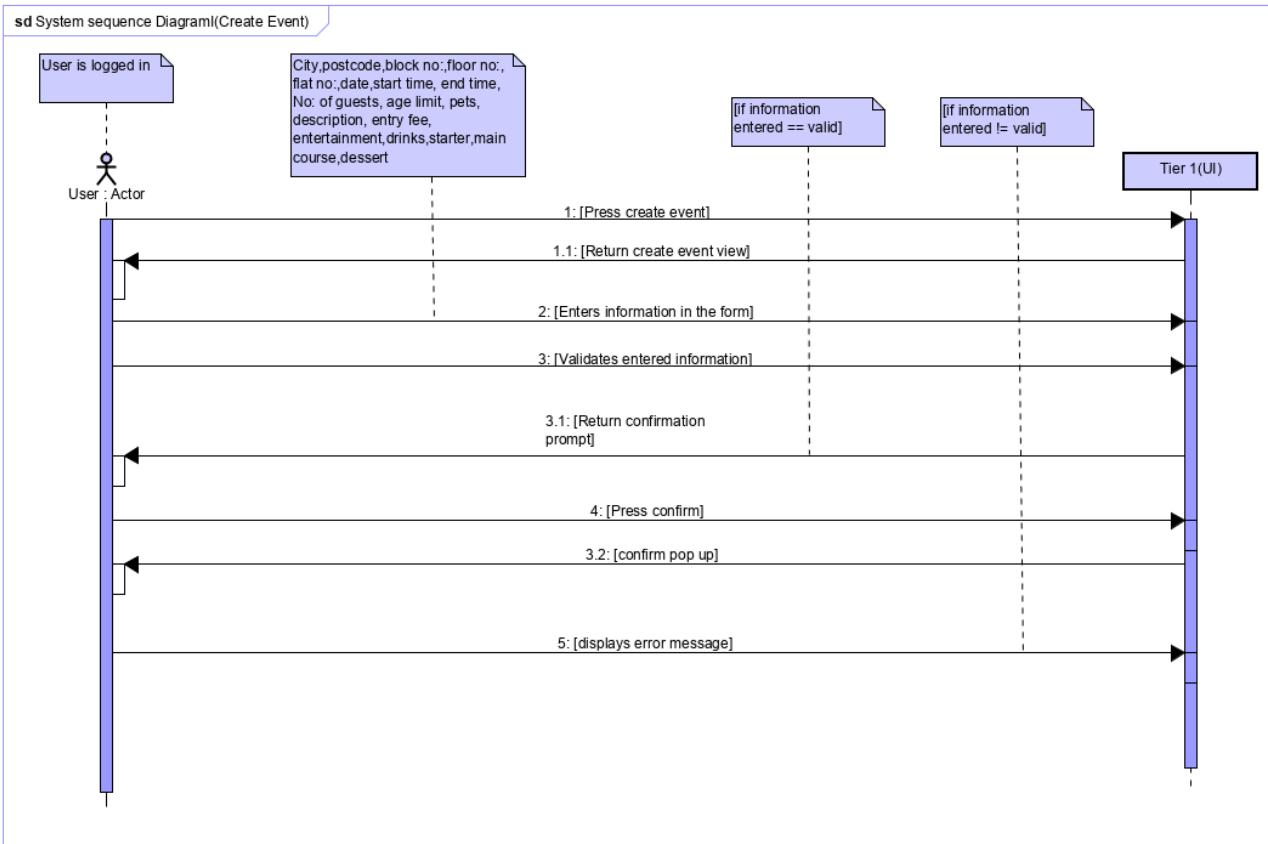
| | |
|-------------------------------------|--|
| Use Case Section | |
| Use Case Name | Manage account |
| Scope | Dining with strangers. |
| Level | User goal. |
| Primary Actor | User |
| Stakeholders and interests | User wants to manage its account to make sure it's up to date with valid information. |
| Preconditions | User is authenticated by the system. |
| Success Guarantee | The user successfully manages its account by either <u>updating</u> information, <u>changing passwords</u> , seeing personal information, <u>deleting account</u> or <u>uploading a picture of myself</u> . With the changes being stored in the system. |
| Main Success Scenario | <ol style="list-style-type: none"> 1. The user selects account settings. 2. System returns account details interface. 3. The user selects edit. 4. System returns an editable version of account details. 5. User changes desired fields. 6. User selects the Save button. 7. System returns confirmation message. 8. User confirms. 9. System updates changes. |
| Extensions | Branch 1- User cancel update account. 1-4 base sequence 5. User selects cancel. 6. System returns confirmation message. 7. User confirms cancel confirmation message. 8. System returns account settings. |
| Special Requirements | |
| Technology and Data variations list | |
| Frequency of occurrence | |
| Open issues | |

Appendix D: Sequence diagrams

Create event sequence diagram

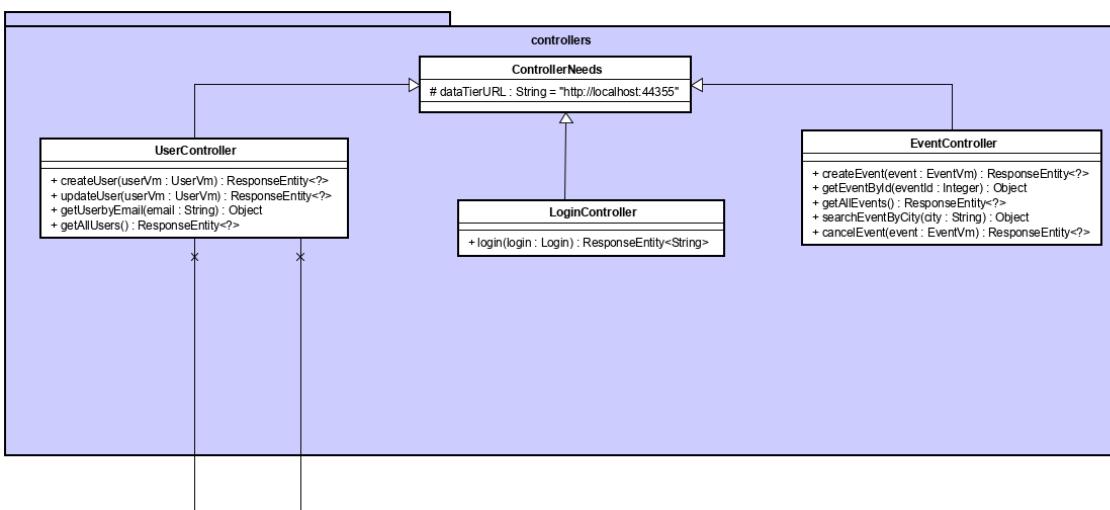


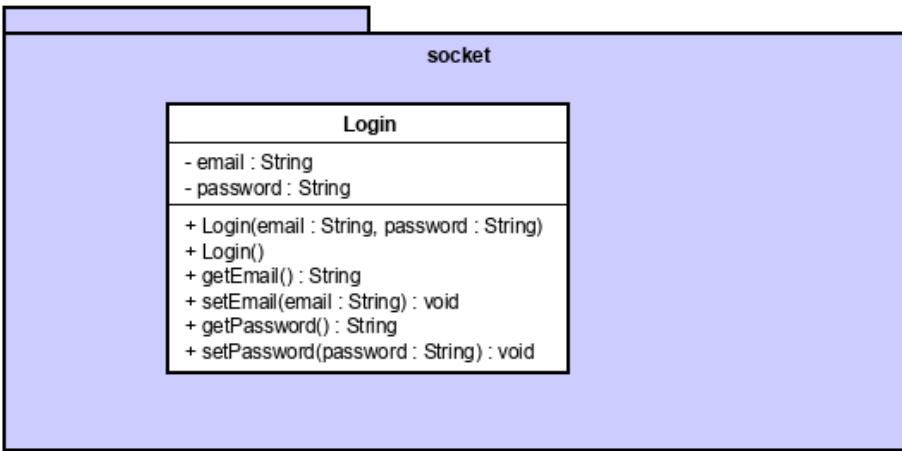
System sequence diagram for Create event

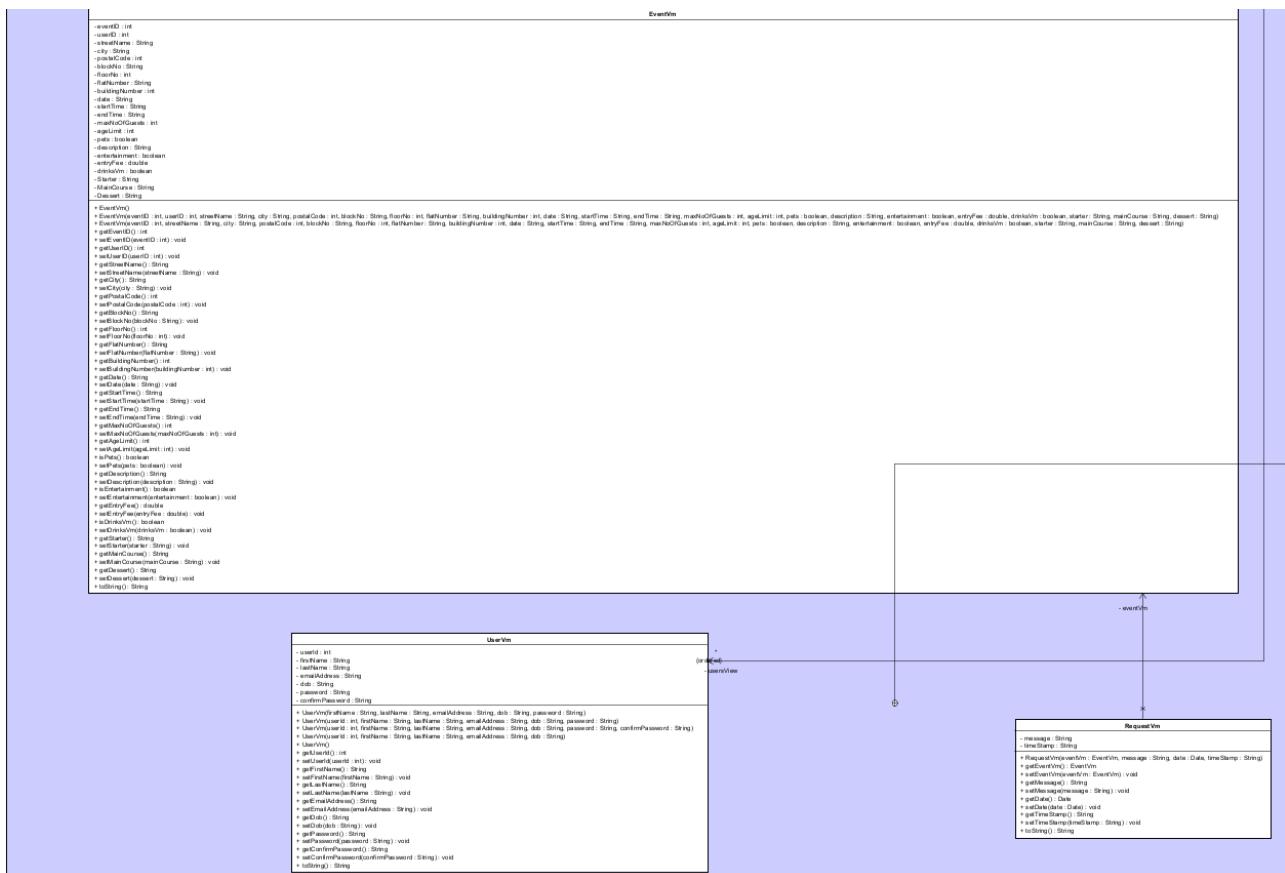


Appendix E: Class diagrams

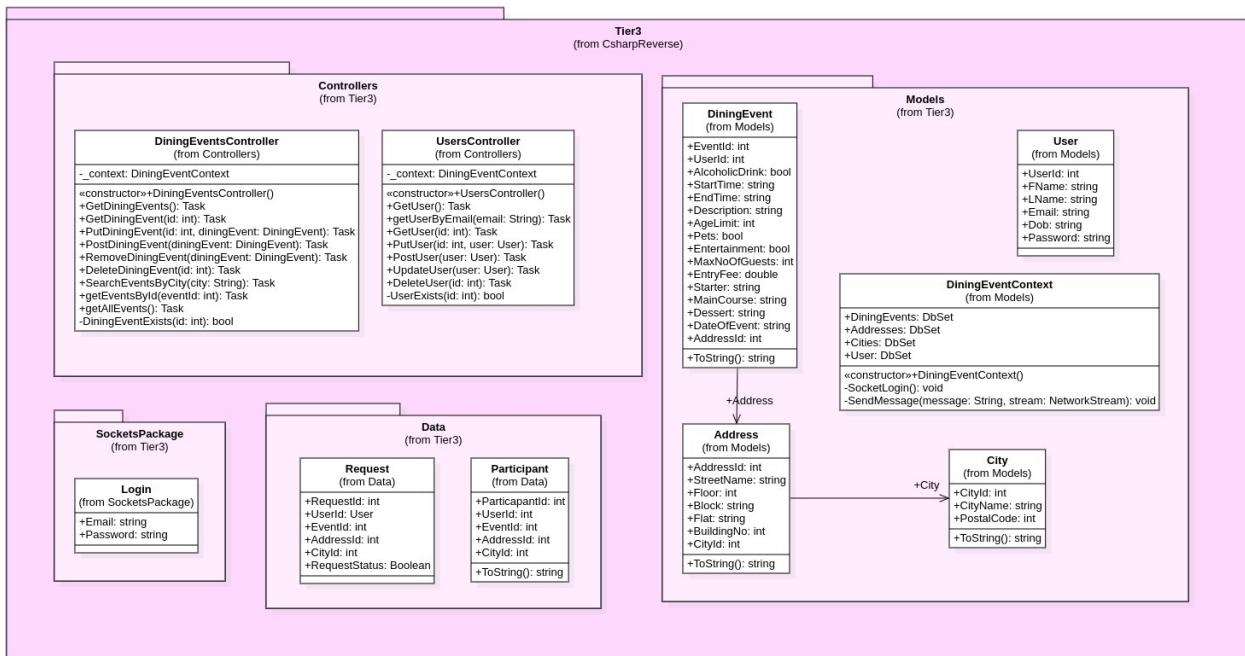
Second tier







Third tier



Appendix F: Java sockets implementation

```
14  @RestController
15  @RequestMapping("startApplication/api")
16  public class LoginController extends ControllerNeeds {
17
18      @RequestMapping(value = "/login", method = RequestMethod.POST)
19      public ResponseEntity<String> login(@RequestBody Login login) {
20          try {
21              Socket socket = new Socket( host: "localhost", port: 4567);
22              InputStream input = socket.getInputStream();
23              OutputStream output = socket.getOutputStream();
24              String json = objectMapper.writeValueAsString(login);
25              System.out.println(json);
26              byte[] toSendBytes = json.getBytes();
27              int toSendLen = toSendBytes.length;
28              byte[] toSendLenBytes = new byte[4];
29              toSendLenBytes[0] = (byte)(toSendLen & 0xff);
30              toSendLenBytes[1] = (byte)((toSendLen >> 8) & 0xff);
31              toSendLenBytes[2] = (byte)((toSendLen >> 16) & 0xff);
32              toSendLenBytes[3] = (byte)((toSendLen >> 24) & 0xff);
33              output.write(toSendLenBytes);
34              output.write(toSendBytes);
35
36              //It will contain "Email not found" or "Password mismatched" or "Login successful"
37              byte[] lenByte = new byte[4];
38              input.read(lenByte, off: 0, len: 4);
39              int len = ((lenByte[3] & 0xff) << 24 | ((lenByte[2] & 0xff) << 16) | ((lenByte[1] & 0xff) << 8) | ((lenByte[0] & 0xff)));
40              byte[] received = new byte[len];
41              input.read(received, off: 0,len);
42
43
44              String responseFromTier3 = new String(received, offset: 0,len);
45              if (responseFromTier3.equals("Email not found") || responseFromTier3.equals("Password mismatched")) {
46                  return ResponseEntity.status(HttpStatus.BAD_REQUEST).body( t: "Try again!");
47
48
49              String responseFromTier3 = new String(received, offset: 0,len);
50              if (responseFromTier3.equals("Email not found") || responseFromTier3.equals("Password mismatched")) {
51                  return ResponseEntity.status(HttpStatus.BAD_REQUEST).body( t: "Try again!");
52
53              } else {
54                  return ResponseEntity.status(HttpStatus.OK).body( t: "Login successful");
55
56          } catch (IOException e) {
57              e.printStackTrace();
58              return ResponseEntity.status(HttpStatus.BAD_REQUEST).body( t: "Try again!");
59
60      }
```

Appendix G: C# listener

```
35     private void SocketLogin()
36     {
37         Console.WriteLine("Starting Server...");
38         IPEndPoint host = Dns.GetHostEntry("localhost");
39         IPAddress ipAddress = host.AddressList[1];
40         IPEndPoint endPoint = new IPEndPoint(ipAddress, port: 4567);
41         try
42         {
43             Socket listener = new Socket(ipAddress.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
44             listener.Bind(endPoint);
45             //number of clients that can connect.
46             listener.Listen(backlog: 100);
47             Console.WriteLine("Server started");
48             Socket client = listener.Accept();
49             Console.WriteLine("Client accepted");
50             byte[] bytes = new byte[1024];
51             int bytesRead = client.Receive(bytes);
52             String messageFromClient = Encoding.ASCII.GetString(bytes, index: 4, count: bytesRead);
53             Console.WriteLine(messageFromClient);
54             Login login = JsonConvert.DeserializeObject<Login>(messageFromClient);
55             Console.WriteLine(login.Email);
56             var userFromDb = User.Single(a => a.Email.Equals(login.Email));
57             if (userFromDb == null)
58             {
59                 string msg = "Email not found";
60                 byte[] bytemsg = Encoding.ASCII.GetBytes(msg);
61                 client.Send(bytemsg);
62             }
63             else if (userFromDb.Password.Equals(login.Password))
64             {
65                 string msg = "Login successful";
66                 byte[] bytemsg = Encoding.ASCII.GetBytes(msg);
67                 client.Send(bytemsg);
68             }
69             else
70             {
71                 string msg = "Password mismatched";
72                 byte[] bytemsg = Encoding.ASCII.GetBytes(msg);
73                 client.Send(bytemsg);
74             }
75         }
76         catch (Exception e)
77         {
78             Console.WriteLine("oopsie doopsie we made a whoopsie" + e.ToString());
79         }
80     }
```

Appendix H: System tests

Test cases – Create account

| Action no. | Action | Reaction |
|------------|--|--|
| 1. | Select register | Verify that the account creation interface is displayed |
| 2. | Fill in: first name, last name, e-mail, date of birth, password. | Verify that the system is recording user inputs. |
| 3. | Select Save button | Verify that passwords must match. Verify that you cannot use the same e-mail twice. Verify that you cannot create an account with empty fields. Verify that you cannot create an account with an invalid e-mail address Verify that the system asks for confirmation. Verify that you're returned to account creation when errors are thrown. |
| 4. | Select Confirm | Verify that the system returns a confirmation when an account has been stored in the system. |
| 5. | Select Cancel | Verify that you're returned to the account creation. |
| 6. | Cancel account creation | Verify that the system asks for confirmation. Verify that the system returns you to the main page. |

| Test step | Test data | Expected result | Actual result | Notes |
|-----------|-----------|-----------------|---------------|-------|
| | | | | |

| | | | | |
|---|--|--|--------|--|
| User inputs: <ul style="list-style-type: none">• First name• Last name• Email• Password• Password repeated• Date of birth | <ul style="list-style-type: none">• John• Norris• jn@jn.com• 123456• 123456• 1/11/2000 | No error message is displayed, all the fields have green outline | Passed | |
| User presses Save button | | Confirmation prompt appears | Passed | |
| User presses Confirm registration button | | User is navigated to Search events page, successful message appears Database entry appears | Passed | |

Test cases – Create event.

| Action no. | Action | Reaction |
|------------|---------------------------|---|
| 1. | Select create event | Verify that the event creation interface is displayed. |
| 2. | Fill the fields with data | Verify that the system is recording user inputs. |
| 3. | Select Create button | Verify that the system asks for confirmation. |
| 4.a. | Select Create | Verify that the event cannot be created with mandatory fields being empty. Verify that you cannot create an event with a given date, if you already have an event scheduled for that date. Verify that you cannot create an event with the date being in the past. Verify that you cannot create an event with maxNoOfGuests being 0 or negative. Verify that the age limit cannot be negative. |

| | | |
|------|---|--|
| 4.b. | Select Cancel | Verify that you're returned to the create event interface. |
| 5. | Await confirmation of event being stored. | Verify that the system confirms the event creation. |
| 6. | Cancel event creation. | Verify that you're returned to the main page. |

| Test step | Test data | Expected result | Actual result | Notes |
|----------------------------|---|---|---------------|----------------------|
| User inputs: | <ul style="list-style-type: none"> • City • Post code • Street • Block number • Floor number • Flat number • Date • Start time • End time • Number of guests • Age limit • Pets • Description • Entry fee • Entertainment • Drinks • Starter • Main course • Dessert | <ul style="list-style-type: none"> • Horsens • 4000 • Happy Feet • 4 • 2 • 1 • 9/4/2019 • 12:30 • 18:30 • 6 • 26 • No • Birthday party • 100 • Yes • Yes • Soup • Pizza • Cake | Passed | |
| User presses Create button | | Confirmation prompt appears | Passed | |
| User presses Create button | | User is navigated to Search events page, successful message appears | Not passed | Rendering page error |
| | | Database entry appears | Passed | |

Test cases – Search event.

| Action no. | Action | Reaction |
|------------|---------------------------|--|
| 1. | Select Search events tab | Verify that the search interface is displayed. |
| 2. | Fill the fields with data | Verify that the system is recording user inputs. |
| 3. | Select Search button | Verify that the system returns list of events. |

| Test step | Test data | Expected result | Actual result | Notes |
|----------------------------|-----------|-------------------------------------|---------------|-------|
| User inputs: • City | • Horsens | | Passed | |
| User presses Create button | | Event list is displayed | Passed | |
| | | Database information is retrieved b | Passed | |

Test cases – View account details.

| Action no. | Action | Reaction |
|------------|--|--|
| 1. | Press login link | Verify that the interface is displayed. |
| 2. | Provide login information in the login interface | Verify that the system is recording user inputs. |
| 3. | Press Login button | Verify if the home page interface is returned. |
| 4. | Select Account settings tab | Verify that the interface is displayed. |

| Test step | Test data | Expected result | Actual result | Notes |
|-----------|-----------|-----------------|---------------|-------|
| | | | | |

| | | | | |
|--|--|-----------------------------------|--------|--|
| User inputs: <ul style="list-style-type: none">• Email• Password | <ul style="list-style-type: none">• jn@jn.com• 123456 | Account details are displayed | Failed | |
| | | Database information is retrieved | Passed | |

Test cases – View event details.

| Action no. | Action | Reaction |
|------------|--|--|
| 1. | Press Search events tab | Verify that the interface is displayed. |
| 2. | Select one of the events from the list | Verify that the system is recording user inputs. The selected event should be shaded |
| 3. | Press the event link | Verify if the event details interface is returned. |

| Test step | Test data | Expected result | Actual result | Notes |
|--|--|-----------------------------------|---------------|-------|
| User inputs: <ul style="list-style-type: none">• Email• Password | <ul style="list-style-type: none">• jn@jn.com• 123456 | Account details are displayed | Failed | |
| | | Database information is retrieved | Passed | |

Test cases – View history – Branch 1 – List of events

| Action no. | Action | Reaction. |
|------------|----------------------|---|
| 1. | Select view history | Verify that the View history interface is displayed. |
| 2. | Select Event History | Verify that the Event History interface is displayed and showing all events by default. |

| | | |
|----|------------------------|---|
| 3. | Change search criteria | Verify that the system returns lists matching the newly assigned search criteria. |
|----|------------------------|---|

| Test step | Test data | Expected result | Actual result | Notes |
|--|-----------|---|---------------|-----------------|
| User select History | | System returns History interface | Passed | |
| User presses View event history button | | Event list is displayed | Passed | |
| User presses the Upcoming tabs | | Events are filtered according to category | Not passed | Not implemented |

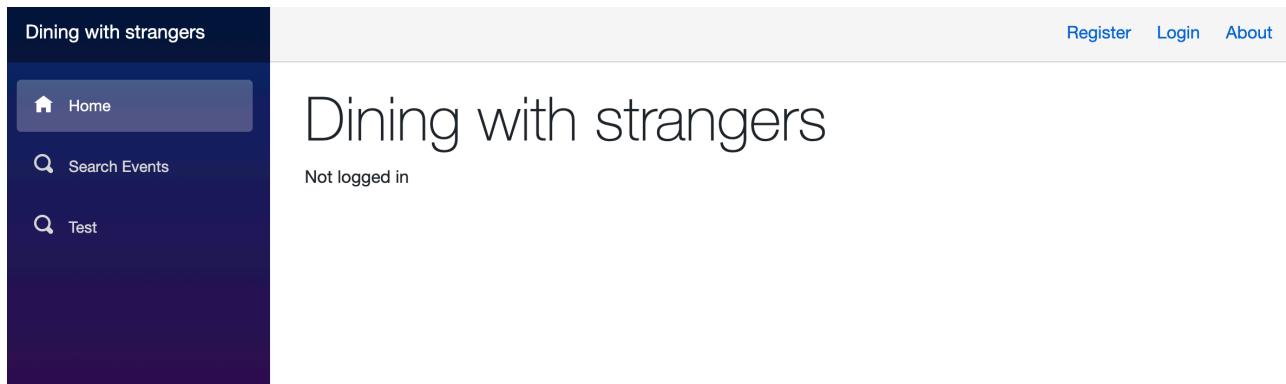
Test Cases – View history – branch 2 – List of requests. (Not implemented)

| Action no. | Action | Reaction. |
|------------|----------------------|--|
| 1. | Select view history | Verify that the View history interface is displayed. |
| 2. | Select Event History | Verify that the request history interface is returned. |
| 3. | Observe. | Verify that the system returns lists with; pending, received and answered request. |

Appendix I: User manual

User manual contains instructions for following views:

1. Account creation
2. Log in
3. Search events
4. View event details
5. Create event
6. View history of your events
7. Edit event
8. Edit account settings



1. Create account - in order to register in the system user needs to complete the following actions:

- Select the Register link in the top right corner.
- Fill out the fields marked with the asterisk sign as required. The following parameters are required for successful registration: first name, last name, email address, password, date of birth.
- Press the Save button on the bottom of the form.
- The confirmation prompt will appear when all of the entered fields are correct, asking the user to confirm if information is correct.
- Notification system display success label after successful account creation.

Exceptions:

System will mark the required fields red if they are not filled out.

System will display error messages if given information does not satisfy requirements for creating user account.

Dining with strangers

Home Search Events Test

Register Login About

John * Required: Enter your first name.

Norris * Required: Enter your last name.

jn@jn.com * Required: Enter your e-mail address.

***** * Required: Enter your password.

***** * Required: Repeat your password.

01/11/2000 * Required: Enter your date of birth.

Confirm that entered information is correct.

Confirm registration Cancel

2. Log in interface

In order to log in to the system user has to:

- Provide valid credentials; email and password.
- Press Login button

Dining with strangers

Home Search Events Test

Login

Email address: email

Password: password

Login

The following interface will be displayed after logging in. User can select actions from the side navigation menu.

Dining with strangers

Hello, d Log out About

Logged in

Home Search Events Test Create Event History Account settings

3. Search events interface.

Search interface allows the user to find events happening in selected city. In order to search events user needs to:

- Enter the city name in the input field.
- Press Search button

The system will return list of upcoming events in chosen city. In order to search events user does not have to be logged in to the system.

The screenshot shows a mobile-style interface for searching events. On the left is a sidebar with links: Home, Search Events (which is active), and Test. The main area has a header "Search events in your area" and a "City" input field. Below is a table titled "Search" with columns Host, Location, Date, Guests, and Description. The table contains five rows of event data, each with a truncated description.

| Host | Location | Date | Guests | Description |
|------|----------|-----------|--------|---|
| 0 | Aarhus | 1/11/1111 | 9 | Non curabitur gravida arcu ac. Interdum velit euismod in pellentesque massa placerat quis. Pharetra et ultrices neque ornare aenean euismod elementum nisi quis. |
| 1 | Horsens | 1/11/1111 | 3 | Tristique senectus et netus et malesuada fames ac turpis. Sed faucibus turpis in eu mi bibendum neque egestas congue. Et netus et malesuada fames ac turpis egestas integer. Tellus rutrum tellus pellentesque eu tincidunt tortor aliquam. |
| 2 | Horsens | 1/11/1111 | 2 | Non curabitur gravida arcu ac. Interdum velit euismod in pellentesque massa placerat quis. Pharetra et ultrices neque ornare aenean euismod elementum nisi quis. |
| 3 | Aarhus | 1/11/1111 | 2 | Non curabitur gravida arcu ac. Interdum velit euismod in pellentesque massa placerat quis. Pharetra et ultrices neque ornare aenean euismod elementum nisi quis. |
| 4 | Aarhus | 1/11/1111 | 4 | Nunc eget lorem dolor sed viverra in ipsum nunc. Mauris in |

4. View event details interface.

After entering city name in the Search events interface system will return list of upcoming events. User can select to view the details of event by clicking on the event from the list. The system will navigate the user to the page where all event details can be seen.

The screenshot shows a detailed view of an event. The sidebar includes Home, Create Event (active), History, and Account settings. The main content is titled "Event details 0". It shows the host (fghjk), location (Horsens), date (December 18), and a description: "Id neque aliquam vestibulum morbi blandit. Dignissim suspendisse in est ante in nibh. Urna cursus eget nunc scelerisque viverra mauris." It also lists the number of guests (4), entry fees (0), age limit (0), and categories like Starter, Main course, Dessert, Pets, Drinks, and Entertainment, each with a checked checkbox. At the bottom is a "Request to join the event" button.

5. Create event interface

In order to create event user has to be logged into the system. The following actions have to be completed:

- Select Create event tab from the menu on the left side of the page.
- System will navigate the user to the interface.
- Provide required information in order to create an event: city, street, post code, building number, date of the event, starting time, ending time, number of guests, pets, drinks, entry fee, entertainment and description (fields block number, flat number, floor number, age limit, starter and desert are optional).
- When all of the fields are correctly filed out, select Create button.
- The confirmation prompt will appear, asking the user to confirm if information is correct.
- Confirm with selecting Create button
- Notification system displays success label after successful event creation.

The screenshot shows the 'Create Event' interface. On the left, a sidebar menu includes 'Home', 'Search Events', 'Create Event' (which is selected), 'History', and 'Account settings'. The main content area is titled 'Create Event'. It has several sections: 'Location' (City, Post code, Street, Block number, Flat number, Floor number), 'Date' (Date, Start, End), 'Guests' (Number of guests, Age limit), 'Description' (Description, Entry fee), and 'Entertainment' (checkbox for pets). Each section contains input fields and validation messages.

Description

Birthday party

* Required: Write a short description of your event

100

Specify entry fee.

Is entertainment included?

Drinks

Are drinks included?

Food

Soup

Enter starter information

Pizza

* Required: Enter main course information

Cake

Enter desert information

Confirm that entered information is correct

Create **Cancel**

6. View history of your events interface

In order to see history of events, user has to be logged into the system.

- Select History tab from the navigation menu on the left side.
- Select View events history button
- System will return list of user's events.

The screenshot shows a user interface for managing events. On the left, there is a dark sidebar with navigation links: Home, Search Events, Create Event, History (which is selected), and Account settings. The main content area has a title 'Your event history'. Below the title is a navigation bar with three tabs: All, Upcoming (which is selected), and Past. A table lists five past events. Each row contains columns for Host (fghjk), Location (Horsens), Date (18/12/2019, 19/12/2019, 20/12/2019, 21/12/2019, 22/12/2019), Number of guests (4, 5, 1, 6, 3), Description (a block of placeholder text), and an 'Edit' button.

| Host | Location | Date | Number of guests | Description | Edit |
|-------|----------|------------|------------------|---|------|
| | | | | | |
| fghjk | Horsens | 18/12/2019 | 4 | Id neque aliquam vestibulum morbi blandit. Dignissim suspendisse in est ante in nibh. Urna cursus eget nunc scelerisque viverra mauris. | |
| fghjk | Horsens | 19/12/2019 | 5 | Non curabitur gravida arcu ac. Interdum velit euismod in pellentesque massa placerat duis ultricies lacus. Pharetra et ultrices neque ornare aenean euismod elementum nisi quis. | |
| fghjk | Horsens | 20/12/2019 | 1 | Sed faucibus turpis in eu mi bibendum neque egestas congue. Et netus et malesuada fames ac turpis egestas integer. Tellus rutrum tellus pellentesque eu tincidunt tortor aliquam. | |
| fghjk | Horsens | 21/12/2019 | 6 | Id neque aliquam vestibulum morbi blandit. Dignissim suspendisse in est ante in nibh. Urna cursus eget nunc scelerisque viverra mauris. | |
| fghjk | Horsens | 22/12/2019 | 3 | Diam donec adipiscing tristique risus nec. Ullamcorper sit amet risus nullam. Congue eu consequat ac felis. Vel risus commodo viverra maecenas accumsan laoreet vel. | |

7. Edit event interface

From the interface for viewing history of events, user has an option to edit upcoming events. In order to make updates to existing upcoming event, user needs to:

- Select the event the user wants to update by clicking the edit icon on the right side of the list. System returns the event interface.
- Presses Edit button.
- Change the information in desired field(s).
- Select Save changes.
- System returns notification for successfully updating user's event.

From the editing event interface user can request to cancel the event by completing following actions:

- Select the event the user wants to cancel by clicking the edit icon on the right side of the list. System returns the event interface.
- Press Cancel event button.
- System returns confirmation prompt, asking for confirming cancellation of the event.
- Press Confirm cancellation button.
- System returns notification message that the selected event has been cancelled.

The figure consists of two vertically stacked screenshots of a web-based event management application. Both screenshots show a sidebar on the left with navigation links: Home, Search Events, Create Event, History, and Account settings. The main area is titled 'Edit your event 0'.

Top Screenshot (Successful Edit):

- Location:** City: Horsens, Street: (highlighted in green), Building number: 0
- Date:** December 18, Event starts at: (highlighted in green), Event ends at: (highlighted in green)
- Description:** Id neque aliquam vestibulum morbi blandit. Dignissim suspendisse in est ante in nibh. Urna cursus eget nunc scelerisque viverra mauris.
- Number of guests:** 4 (highlighted in green)
- Entry fees:** 0
- Age limit:** 0
- Starter:** Starter
- Main course:** Main course
- Dessert:** Dessert
- Pets:** ✓
- Drinks:** ✓
- Entertainment:** ✓

Bottom Screenshot (Attempt to Cancel):

- Location:** City: Horsens, Street: (highlighted in red), Building number: 0
- Date:** December 18, Event starts at: (highlighted in red), Event ends at: (highlighted in red)
- Description:** Id neque aliquam vestibulum morbi blandit. Dignissim suspendisse in est ante in nibh. Urna cursus eget nunc scelerisque viverra mauris.
- Number of guests:** 4
- Entry fees:** 0
- Age limit:** 0
- Starter:** Starter
- Main course:** Main course

Both screenshots show 'Edit' and 'Cancel event' buttons at the bottom.

8. Edit account settings interface

From this interface user can update the account settings. In order to update account credentials user needs to:

- Select the Account settings tab
- Presses Edit button.
- Change the information in desired field(s).
- Select Save changes.
- System returns notification for successfully updating user's credentials.

