Acceptance Test Plan

Project Code Defenders - Robo Tournament

Team Codebenders

Agenda

- Our Team
- Project Vision and Requirements
- Performing Acceptance Tests
- Acceptance Tests Details

Our Team

Frontend

Product Owner

Fanny Delnondedieu

Scrum Master

Dominik Brdar

Testers

Hrvoje Rom

Fabio Patella

Backend

Simone Mezzaro

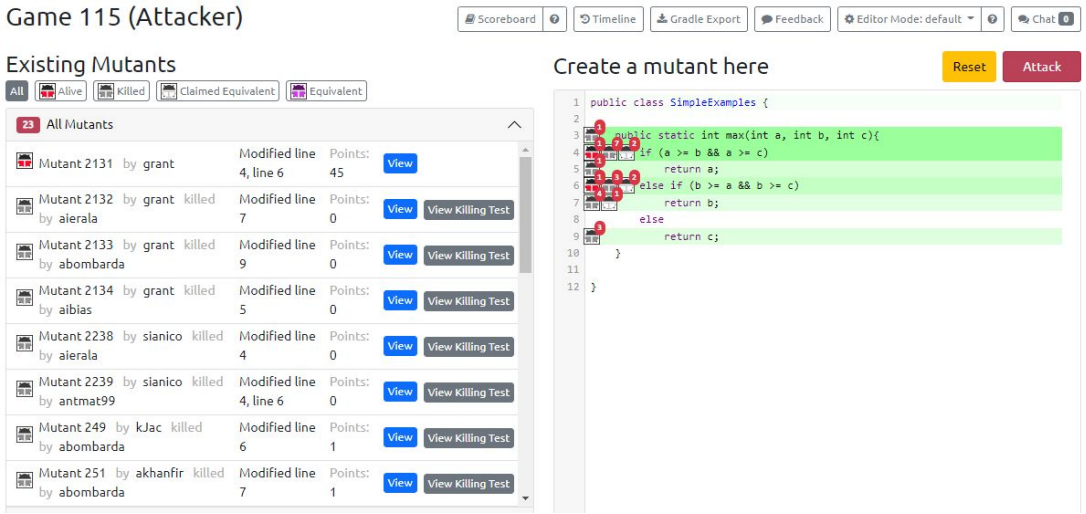
Riccardo Nava

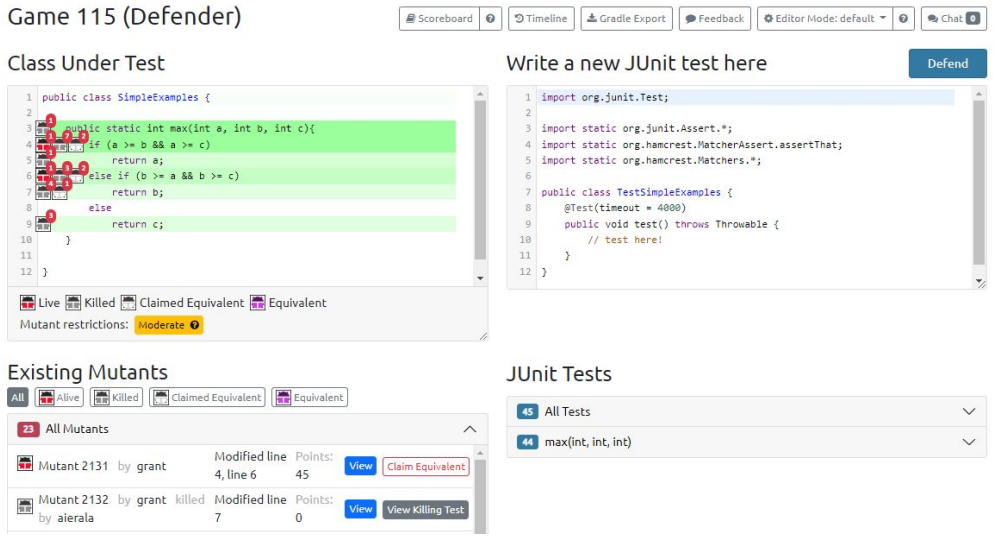
Andrea Restelli

Project vision

Code Defenders

- Software quality and testing are at the heart of software engineering, but they may not always get enough attention from software engineering education.
- CodeDefenders (web game) proposes the use of **gamification** to teach **mutation testing** and to strengthen code writing and testing skills.
- The game supports **team play and competition** by having Attackers - Defenders teams whose goal is to inject errors into code or write unit tests to catch them.
- The **“CodeDefenders: RoboTournament”** project aims at enriching the game by adding support for students tournaments and games against bots.





- For more details refer to Project Plan document

Project requirements

- Implement a **tournament application**. This application must use CodeDefenders as a remote service (through APIs) and must include at least two tournaments modalities.
- Design and implement a set of **OpenAPIs for CodeDefenders** which can be used from the tournament application to manage games and players.
- Implement a **load balancing** mechanism which allows the tournament application to communicate with **multiple CodeDefenders servers** and to always create games on the less loaded server.
- Implement a **streaming** component which allows users to follow in progress games live. This component can optionally include an “overall tournament view” showing schedule, standings and other information for each tournament.
- Design and implement a set of **APIs** which allows users to train **bots** over past games data and to let those bots play CodeDefenders.
- Please refer to document *Requirements Definition* for more details

Performing Acceptance Tests

- We listed all the actions that can be performed in our application
  - Each action maps one or more requirements by the customer and is covered by one specific acceptance test
  - Each test in this document has ID, name and description (what action is it testing), link to User Story containing mapped project requirement, prerequisite (if any), procedure (instructions in steps how to carry out the test), and how to evaluate results (pass criteria, if not met - test failed)
- We plan to go through all the tests together with our customer to verify that they are successful and that the behavior of the application is the one required by him
- Finally, the customer will express whether the product we implemented meets all his needs or not
- Acceptance tests coverage of the user stories is shown in the next slides

Tests mapped onto User Stories

CDF-32 Login/Register

Test-1 Registration
Test-2.1 Successful Login
Test-2.2 Rejected Login

CDF-41 Display Tournaments Info

Test-3 Display Tournaments Information

CDF-35 Team Creation

Test-4 Create a Team

CDF-54 Team Management

Test-5 Leave the Team
Test-6 Kick Members Out of the Team
Test-7 Promote Team Member as Leader
Test-8 Invite Players to the Team

CDF-37 Join Team

Test-9 Join an Open Team
Test-10.1 Accept Received Invitations
Test-10.2 Decline Received Invitations

CDF-33 Create Tournament

Test-11 Create Tournament
Test-20 Upload a Class
Test-21 Choose a Class

CDF-34 Join Tournament

Test-12 Join Tournament

Tests mapped onto User Stories

CDF-36 Starting Games

Test-13 Tournament is Started
Test-14 Games are Split in Phases
Test-15 Users can Play Games

CDF-38 Leave Game and Game End

Test-16 Return to Tournament App

CDF-39 View Game Stream

Test-17 Join a Game Streaming

CDF-40 Notifications of Game Stream Updates

Test-18 Receive Game Streaming Updates

CDF-31 Load Balancing

Test-19 Load Balancing
Test-22.1 Register a CD Server Instance
Test-22.2 Update a CD Server Instance
Test-22.3 Remove a CD Server Instance

CDF-43 Bots can Play

Test-23 Bots can Play a Game

CDF-44 Bots can be Trained

Test-24 Bots can be Trained

Test 1 - Registration

Description: a non-registered user can register to the Tournament Application by filling a form asking username, email and password

Tested user story: CDF-32

Pass criteria: the user receives a “registration successful” message and is redirected to the login

Test procedure:

- On the homepage press “SignIn” button
- Select “Create Account”
- Fill the form with username, email and password repeated two times
  - username must be from 3 to 20 characters long, can contain only uppercase letters, lowercase letters and digits, can NOT start with a digit. It also must not been already taken by another user
  - email must be valid and not already taken by another user
  - password must be from 8 to 20 characters long, can contain only uppercase letters, lowercase letters and digits, must contain at least one uppercase, one lowercase and one digit
  - the two passwords must be identical
- Press “Sign up” button
  - if you haven’t respected the criteria listed above you’ll receive an error message. Fill the form again following the instructions in the error

Test 2.1 - Successful Login

Description: a non-authenticated but already registered user can login to the Tournament Application by filling a form with username and password

Tested user story: CDF-32

Prerequisite: the user has already registered an account

Pass criteria: the user is redirected to the home page and can see its username in the “Welcome” message in the page header

Test procedure:

- On the homepage press “SignIn” button
- Fill the form with username and password of an already registered account
- Press “Sign in” button

Test 2.2 - Rejected Login

Description: a non-authenticated non-registered user can not login to the Tournament Application

Tested user story: CDF-32

Prerequisite: the user has not registered an account yet

Pass criteria: an error message stating that the credentials are wrong is displayed to the user

Test procedure:

- On the homepage press “SignIn” button
- Fill the form with username and password of a non-registered account
- Press “Sign in” button

Test 3 - Display Tournaments Information

Description: any user (authenticated or not) can view a list of all the tournaments and expand view of any tournament entry to see additional info. (Initially, simplified list is displayed so that users can browse tournaments more easily)

Tested user story: CDF-41

Pass Criteria: the list of tournaments is visible on the homepage, additional info is hidden at first and visible after user clicks on tournament entry (more details in test procedure)

Test procedure:

- Access the homepage by clicking “Home” button
- The list of tournaments is displayed with their status, name, team size and a green/red button if joinable/not joinable
- Click on a specific tournament
- Names of teams participating in the tournament are displayed
- Winner is displayed if any
- List of scheduled matches is shown if tournament is not in *Teams_Joining* or *Selecting_classes* status
- For each match, names of the attacking and defending team, status, winner and play button (if user is inside the match) and livescore button (if game is in progress) are displayed

Test 4 - Create a Team

Description: an authenticated user can create a new team

Tested user story: CDF-35

Prerequisite: the user has logged in and is not already in a team

Pass Criteria: A message “team successfully created ” is displayed

Test procedure:

- On the home page click on “Create Team” button
- A form with fields for the name of the team, team size and type of team (open/close) are displayed
- Fill the fields and click on the button “Create Team”
  - the name must be 255 char maximum
  - the name must not be already taken by another team

Test 5 - Leave the Team

Description: Users that are members (not leader) of a team can leave the team.

Tested user story: CDF-54

Prerequisite: the user is already in a team and is not the leader. The team is not in a tournament.

Pass criteria: “You are not in any team” message is displayed in “Manage Teams” section

Test procedure:

- On the homepage press “Team Management” button
- Press “Leave” button next to your username on the table of team members.
- Click on “Manage Teams” button

Test 6 - Kick Members Out of the Team

Description: Users that are leaders of a team can kick members out of the team.

Tested user story: CDF-54

Prerequisite: the current user is the leader of his team. The user that we want to kick is in the same team as the current user.

Pass criteria: “You are not in any team” message is displayed in “Manage Teams” section for the kicked out user

Test procedure:

- On the homepage press “Team Management” button
- Press “Kick from the team” button next to the username of the targeted user on the table of team members
- Log in with the kicked out user
- Click on “Manage Teams” button

Test 7 - Promote Team Member as Leader

Description: Users that are leaders of a team can promote members as leader of the team.

Tested user story: CDF-54

Prerequisite: the current user is the leader of his team. The user that we want to promote is in the same team as the current user.

Pass criteria: in the table of team members the role of the promoted user is now “LEADER” and the role of the previous leader user is now “MEMBER”

Test procedure:

- On the homepage press “Team Management” button
- Press “Promote to leader” button next to the username of the target user on the table of team members.

Test 8 - Invite Players to the Team

Description: Users that are leaders of a team can invite new members into the team.

Tested user story: CDF-54

Prerequisite: the current user is the leader of his team.

Pass criteria: The target user can see an invitation to the team in the invitations list

Test procedure:

- On the homepage press the “Team Management” button
- Press the “Invite players” button
- Press the “Send invitation” button on the row of the user to invite.
- Acquire a session for the target user
- On the homepage press the “Join Team” button
- Press the “Check Invitations” button

Test 9 - Join an Open Team

Description: Authenticated users can join an open team

Tested user story: CDF-37

Prerequisite: The user is not already part of a team; the target team is not full or playing in a tournament

Pass criteria: The Team Management page shows that the user is a member of the target team

Test procedure:

- On the homepage press the “Join team” button
- Press the button with the name of the target team
- Press the “Join this team” button and press “OK” in the alert messages
- Press the “Manage Teams” button on the sidebar

Test 10.1 - Accept Received Invitations

Description: Authenticated users can accept invitations to join a team.

Tested user story: CDF-37

Prerequisite: The user has a pending invitation; the user is not already part of a team; the target team is not full or playing in a tournament

Pass criteria: The invitation list doesn't contain the target invitation anymore; the Team Management page shows that the user is a member of the team that sent the invitation which user accepted

Test procedure:

- On the homepage press the "Join Team" button
- Press the "Check Invitations" button
- Press the "Accept" button on the target invitation
- Press the "Manage Teams" button on the sidebar

Test 10.2 - Decline Received Invitations

Description: Authenticated users can decline invitations to join a team.

Tested user story: CDF-37

Prerequisite: The user has a pending invitation

Pass criteria: The invitation list doesn't contain the target invitation anymore

Test procedure:

- On the homepage press the "Join Team" button
- Press the "Check Invitations" button
- Press the "Decline" button on the target invitation

Test 11 - Create Tournament

Description: an authenticated user can create a new tournament.

Tested user story: CDF-33

Prerequisite: the current user is logged in.

Pass Criteria: “Tournament successfully created” message is displayed. The newly created tournament is visible in the Home page.

Test procedure:

- On the side menu press “Tournament Creation” button.
- Fill the form with the name of the new tournament, the type (Knockout or League), the size of the teams and the number of teams.
  - the name must be < 255 characters long
  - there must be no other active tournament with the same name
  - in case of Knockout tournaments, only powers of two are accepted as the number of teams participating
- Press “Create tournament” button.
- “Tournament successfully created” message is displayed.
- Return to Home page and the newly created tournament should be visible in the list of all the tournaments.

Test 12 - Join Tournament

Description: an authenticated user can join an existing tournament

Tested user story: CDF-34

Prerequisite: the current user is logged in and he is the leader of his team.

Pass Criteria: “Successfully joined” message is displayed

Test procedure:

- Access the homepage by clicking “Home” button
- The list of tournaments is displayed with their status, name, team size and a green/red button if joinable/not joinable
- Click on a green “Join” button to join a specific tournament
- A confirmation message is displayed with the name of the user team and a “Join Tournament” button
- Click on the “Join Tournament” button

Test 13 - Tournament is Started

Description: When all the teams have joined a tournament it is started.

Tested user story: CDF-36

Prerequisite: a tournament created with all the teams except one that already joined.

An authenticated user currently in a team not already participating in a tournament.

Pass criteria: the tournament is started and the phase of the tournament moves to SELECTING_CLASSES, the first phase of the tournament.

Test procedure:

- In the Home page press “Join” button on the tournament with one slot left.
- Look at the Status of the tournament being changed to “SELECTING_CLASSES”.

Test 14 - Games are Split in Phases

Description: games are split in three phases, whose duration is set in the application settings. The succession of the different phases is shown by a timer on top of the screen during the game. Depending on the phase, some actions are disallowed.

Tested user story: CDF-36

Prerequisite: an authenticated user involved in an ongoing tournament.
A game about to start.

Pass criteria: in the first X minutes of the game, as shown by the timer on top of the screen, any action is allowed. In the following Y minutes, the upload of new tests and mutants is blocked. In the following Z minutes, also equivalence claims are disabled.

Test procedure:

- On the home page press on the entry corresponding to a tournament the authenticated user is currently involved in.
- Choose the game the user is supposed to be playing. Press “Play” button.
- Look at the timer on top of the window.
- Play the game while considering the timer.

Test 15 - Users can Play Games

Description: an authenticated user can play a game his team is involved in.

Tested user story: CDF-36

Prerequisite: an authenticated user involved in an ongoing tournament.
A game in progress.

Pass criteria: the user is redirected (inside an iframe) to the CodeDefenders instance hosting that game and can play.

Test procedure:

- In the Home page, press on the tournament the player's team is currently in to see the details of the ongoing games.
- Click on the "Play" button beside the ongoing match the player's team is currently playing.

Test 16 - Return to Tournament App

Description: A user playing a game can anytime return to the Tournament App.

Tested user story: CDF-38

Prerequisite: a game in progress.

An authenticated user playing the game.

Pass criteria: the user is redirected to the Tournament App Home page still being logged in.

Test procedure:

- On top left of the screen, press “Back To Tournament Application” button.

Test 17 - Join a Game Streaming

Description: any user (authenticated or not) can join the streaming of an ongoing game and see the current score

Tested user story: CDF-39

Prerequisite: there must be an ongoing game to be watched

Pass criteria: the user can see the streaming page of the game with the current scores of the teams

Test procedure:

- On the homepage select an IN_PROGRESS tournament: a list of games will be displayed
- Choose a game in phase one, two or three and press the “Live Score” button

Test 18 - Receive Game Streaming Updates

Description: any user (authenticated or not) receives live updates (with at most 10 seconds delay) while watching the streaming of a game

Tested user story: CDF-40

Prerequisite: there must be an ongoing game to be watched

Pass criteria: the user can see messages describing the game events and updated scores within 10 seconds

Test procedure:

- This test requires two users: user1 must be in the CodeDefenders page to play an ongoing game; user2 must be in the streaming page of the same game
- Create a valid mutant or a test using user1 and submit it pressing the “Attack” or “Defend” button on CodeDefenders
- Wait at most 10 seconds on the streaming page with user2 to receive updated scores and a notification of mutant or test creation event

Test 19 - Load Balancing

Description: the tournament application always creates new games on the CodeDefenders server instance with the minimum number of ongoing games

Tested user story: CDF-31

Prerequisite: two empty CodeDefenders instanced with no ongoing games registered on the tournament application

Pass criteria: users playing on different games can see different URLs while they are on the playing page in CodeDefenders

Test procedure:

- Create a tournament with 4 teams
- Join the tournament with 4 teams so that it starts and creates two games
- Log in with a user belonging to a team competing in the first game
- On the homepage select the created tournament and press the “Play” button of the first game
- Note the URL of the CodeDefenders page reached in this way
- Repeat the process with a user belonging to a team competing in the second game and checked if the URL of the paged reached in this way is different from the previous one

Test 20 - Upload a Class

Description: any authenticated user can upload classes to play on during tournament games.

Tested user story: CDF-33

Prerequisite: a file .java containing the class to be uploaded.

The user is authenticated.

Pass criteria: the class is uploaded correctly (confirmation message is shown) and can be chosen in Round Choice section.

Test procedure:

- On the home page press “Tournament Creation” section
- On the right, in the “Upload classes” section, press “Select” button
- Choose a file .java containing the class to be uploaded and select it
- Press the “Upload” button

Test 21 - Choose a Class

Description: a user who has created a tournament can choose, for each round of the tournament, the class the matches will be played on.

Tested user story: CDF-33

Prerequisite: at least one class available to choose from.

The user is authenticated and has created a tournament not started yet.

Pass criteria: “Class successfully selected” message is displayed and the choice of the class for that round is updated.

Test procedure:

- On the home page press on the entry corresponding to a tournament created by the user logged and not started yet.
- From the dropdown menu, select the round to choose the class for.
- From the other dropdown menu, select the class among those available.
- Press “Select” button to conclude the choice.

Test 22.1 - Register a CD Server Instance

Description: the admin user can register a new CD server instance with a given address and token

Tested user story: CDF-31

Prerequisite: a new running CodeDefenders server, its address and one of its admin tokens

Pass criteria: user receives a response with code 200 from the server

Test procedure:

- On the login page log in with admin credentials: default username is “admin” and default password is “admin”, if you didn’t change them
- From the browser console run:

```
fetch('<TA_server_address>/admin/api/cd-server/register', {
  method: 'POST',
  credentials: 'include',
  body: JSON.stringify({
    "address": <CD_server_address>,
    "adminToken": <CD_token>
  }),
  headers: {
    'Content-type': 'application/json; charset=UTF-8'
  }
})
```

where *<TA_server_address>* is the address of the backend server of the tournament application, *<CD_server_address>* is the address of the already registered CodeDefenders server and *<CD_token>* is the admin token of that server

Test 22.2 - Update a CD Server Instance

Description: the admin user can update the token of a registered CD server instance with a given address

Tested user story: CDF-31

Prerequisite: a new admin token for an already registered CodeDefenders server

Pass criteria: user receives a response with code 200 from the server

Test procedure:

- On the login page log in with admin credentials: default username is “admin” and default password is “admin”, if you didn’t change them
- From the browser console run:

```
fetch('<TA_server_address>/admin/api/cd-server/update', {
  method: 'POST',
  credentials: 'include',
  body: JSON.stringify({
    "address": <CD_server_address>,
    "adminToken": <CD_token>
  }),
  headers: {
    'Content-type': 'application/json; charset=UTF-8'
  }
})
```

where *<TA_server_address>* is the address of the backend server of the tournament application, *<CD_server_address>* is the address of the already registered CodeDefenders server and *<CD_token>* is the new admin token of that server

Test 22.3 - Remove a CD Server Instance

Description: the admin user can remove a registered CD server instance with a given address

Tested user story: CDF-31

Prerequisite: an already registered CodeDefenders server

Pass criteria: user receives a response with code 200 from the server

Test procedure:

- On the login page log in with admin credentials: default username is “admin” and default password is “admin”, if you didn’t change them
- From the browser console run:

```
fetch('<TA_server_address>/admin/api/cd-server/delete', {
  method: 'POST',
  credentials: 'include',
  body: JSON.stringify({
    "address": <CD_server_address>
  }),
  headers: {
    'Content-type': 'application/json; charset=UTF-8'
  }
})
```

where *<TA_server_address>* is the address of the backend server of the tournament application and *<CD_server_address>* is the address of the already registered CodeDefenders server

Test 23 - Bots can Play a Game

Description: A user can obtain an identification token to authenticate his bot, which can then play a CodeDefenders game by means of APIs

Tested user story: CDF-43

Prerequisite: the user is logged in

Pass criteria: the bot can play a game from start to finish with access to the same data that frontend users have

Test procedure:

- The user finds the bot API token on their profile page
- The user registers to any game, taking note of the game ID
- The user programs a bot (outside of the scope of this test), gives it the game ID and the API token, and the bot can play the game.
  The APIs allow the bot to (assuming compatible game state and player role):
  - Get statistics for a user
  - Get the current game's settings
  - Get the current game's informations (class, state, scores, list of mutants and tests with the respective informations)
  - Upload a test
  - Upload a mutant
  - Claim all mutants on a line as equivalent
  - Get the list of pending equivalences
  - Resolve an equivalence (accept or counter with test)

Test 24 - Bots can be Trained

Description: A user can obtain historical data from previous games by calling an API in order to train a bot.

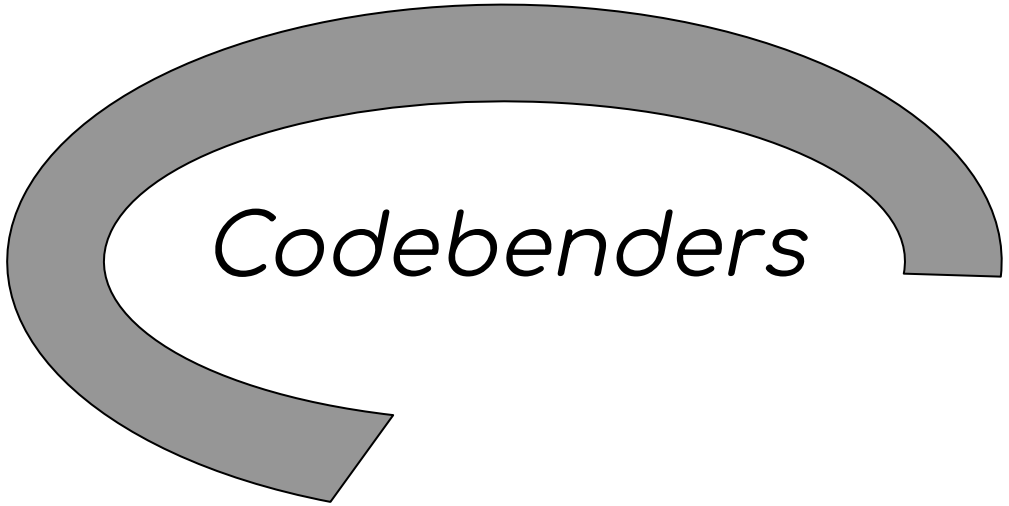
Tested user story: CDF-44

Prerequisite: the user has a valid API token

Pass criteria: the user obtains the requested data

Test procedure:

- The user calls the API endpoint authenticating with the API token to get the historical data



contact info:

fanny.delnondedieu@fer.hr
dominik.brdar@fer.hr
hrvoje.rom@fer.hr
simone.mezzaro@mail.polimi.it
fabio.patella@mail.polimi.it
andrea2.restelli@mail.polimi.it