Confy

Object Design

1.0

10.01.2021

Ali Albaali

Uğur Kaya Küçükkaptan

Bekir Malik Özkuran

Atila Ersun

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc436772639)

[1.1. Object Design Trade-offs 1](#_Toc436772640)

[1.2. Interface Documentation Guidelines 1](#_Toc436772641)

[1.3. Definitions, Acronyms, and Abbreviations 1](#_Toc436772642)

[1.4. References 1](#_Toc436772643)

[2. Packages 1](#_Toc436772644)

[3. Class Interfaces 1](#_Toc436772645)

OBJECT DESIGN DOCUMENT

Object Design Document (ODD) describes object design trade-offs made by developers, guidelines they followed for subsystem interfaces, the decomposition of subsystems into packages and classes, and the class interfaces. The ODD is **used** to exchange interface information among teams and **as a reference during testing**. The audience for the ODD includes system architects (i.e., the developers who participate in the system design), developers who implement each subsystem, and testers.

Among three approaches to generate ODD, we follow “**ODD embedded into source code**” approach in SE301, since the other methods create many redundancies, inconsistencies.

The initial version of the ODD can be written soon after the subsystem decomposition is stable. Both packages and class interfaces can be generated from source code (comments!) by using a tool, which is named Javadoc. Keeping material for the ODD with the source code enables the developers to maintain consistency more easily and rapidly.

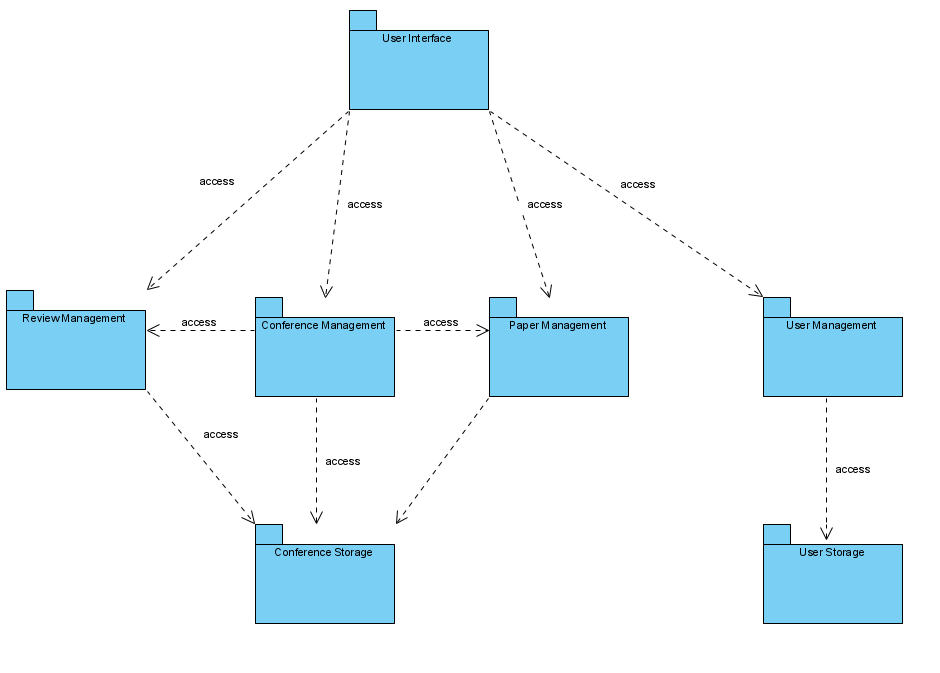
# Introduction

Confy is a programmed conference management application. It is an application where people can create and enter conferences to get their articles and conference topic papers get reviewed and approved. Conferences created start at a given date and after the given date is over. Before the conference begins, the conference can be edited as the conference creator wants it. Conference creators can invite multiple people to the conferences and assign them the role of the reviewer. The duty of reviewers is to help conference creator review and discuss the document sent by the author. The articles that get a review score above the set threshold of the conference gets approved. Users must be registered in order to use Confy services.

Objective of this system is to provide and create an application which satisfies user requests and provide every function mentioned and promised in the RAD document. Application must be simple to use, and not complex to understand. Therefore the promised

system is as simple as it gets. The application will not be usable for unregistered users, as stated before. After creating an account and logging in, registered users will be able to use Confy application and every service and functionality it provides. Information of users will be kept in firebase database in private and will be kept private.

The subsystem decomposition from which the systems and objects are based in this document is summarized in the figure below:



## Object Design Trade-offs

During the object design of the project, we had to make some trade-offs.

We used firebase methods during authentication, such as change password and user functions.

A trade-off was between memory space and response time of the application. Priority was on memory space since the conferences can use some amounts memory to store content that is uploaded.

Another trade-off was about delivery time and functionality of the application. Since we had a already given delivery time we had to put as much as function we can until that time period arrived so we prioritized the delivery time.

## Interface Documentation Guidelines

In order to create better communication between developers during object design we developed some guidelines for interface documentation. We used singular nouns or phrases for class names that included activity or fragment at the end of the name to indicate weather the class is an activity or fragment and every word in the name started with a capital letter. All the objects created were derived from the class diagram and use cases in the RAD. When an exception is caught, the error is displayed to the user.

## Definitions, Acronyms, and Abbreviations

* ***SDD:*** System Design Document
* ***Confy:*** A conference management application which users can upload and review articles and get an approval grade relevant to the conference topic.
* ***Visitor:*** A user who is not registered into system.
* ***Java:*** Java is a programming language.
* ***Firebase Realtime Database:*** A cloud-hosted database provided by Google which allows to store and sync of data.
* ***Subsystem****:* Subsystem is a collection of classes of the system that are closely related to each other.
* ***Article:*** A scientifical writing document that is relevant to the topic of conference that is created.
* ***Conference:*** A meeting of several reviewers that came together to discuss and review an article that is relevant to the conference created.
* ***Reviewers:*** A role during the conference that has responsibility to review and assign grade to the article that is assigned to the conference.

## References

<https://www.exordo.com/>

# Packages

* User Interface Packages

**Organization**:

This package contains /res document of project

/drawable/: These files contain .xml, .png, .jpg files for needed icons and ui elements.

/layout/: These files got the information of user interface templates.

/mipmap/: This file has the launcher icons.

/navigation/: This file got the information about navigation between templates.

/values/: This file has the information about most used strings id, id of colors, themes of using interfaces.

**Overview**

These packages make communication methods of app to user, also it sends user changes to app. And data in the models reach to the user through these files.

**Dependencies:**

User interface package has independence from all other packages.

**Usage:**

User interface package got the all graphical interface for the application and it has the all-design layouts for app. Also it has the images, icons for the application. By that way this package become bridge to activity and fragments of applications to user.

* Conference Management Package:

**Organization:**

These packages contain files of “com.confy.app.conference” documents fragments.

**Overview:**

Conference Management package do the changes on the conference.

**Dependencies:**

Conference management package got the dependencies with user management and paper management packages. It takes the information about user and papers info from these packages

**Usage:**

It takes the actions and information from papers and users packages do the required operation. Then sending all of it to the Conference Storage.

* Paper Management Package

**Organization:**

These packages contain files of “com.confy.app.paper” documents fragments.

**Overview:**

Paper Management package do the changes on the papers.

**Dependencies:**

Paper management package got the dependencies with conference management and user management packages.

**Usage:**

It takes author information from the user management and with the other information about to papers sent to the conference storages to save it.

* Review Management Package

**Organization:**

These packages contain files of “com.confy.app” documents fragments.

**Overview:**

Review Management package do the changes on the papers.

**Dependencies:**

Review management package got the dependencies with conference management and paper management packages.

**Usage:**

It takes paper information from the paper management and took to grade information from user interface and send it to the conference storage to save it.

* User Management Package

**Overview:**

It is responsible for handling user’s information activities by using Firebase Database.

**Usage:**

It is used to perform user profile actions such as edit profile, change password and anything regards user’s information.

**Dependencies:**

It depends on the Account Management Package as a source for retrieving and updating user information.

* User Storage Package

**Overview:**

Files related in this package does not exist in application and they are held inside the Firebase Database. It holds the data of users services such as Login, Register etc.

**Usage:**

It is used to store user information inside the Firebase Database.

**Dependencies:**

It depends on the User Management Package as a source for retrieving and updating user information.

* Conference Storage Package

**Overview:**

Files related in this package does not exist in application and they are held inside the Firebase Database. It holds the data of conferences and reviews services such as grade request and edit conference.

**Usage:**

It is used to store conference information inside the Firebase Database.

**Dependencies:**

It depends on the Conference Management Package and Review Management Package as a source for retrieving and updating conference information. It also has a relation with paper storage system which is used for Paper Management Database storage.

* Paper Storage Package

**Overview:**

Files related in this package does not exist in application and they are held inside the Firebase Database. It holds the data of papers that are reviewed in the conference reviews and shares some data with conference storage package.

**Usage:**

It is used to store paper information inside the Firebase Database.

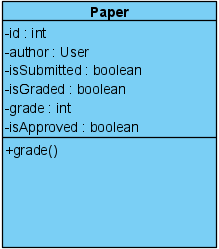
**Dependencies:**

It depends on the Paper Management Package as a source for retrieving and updating conference information. It also has a relation with conference storage system which it has a relation by holding the data of paper of conferences.

# Class Interfaces

Describes the classes and their public interfaces. This includes an overview of each class, its dependencies with other classes and packages, its public attributes, operations, and the exceptions they can raise.

**3.1 Paper Class**



Class Name: Paper

**3.1.1 Attributes**

* + *-author: user*

Author is value of type user and a private attribute. It shows the author of the paper.

* + -id : int

Id is an integer type value, it is an identification number, it is unique for every post object. It is a private attribute. It distinguishes every post object created from each other.

* + *-isSubmitted : boolean*

isSubmitted is one of the papers information of type boolean and it is a private attribute. It provides to see paper is submitted or not.

* + -isGraded : boolean

IsGraded is one of the papers information of type boolean and it is a private attribute. It provides to see paper is graded or not.

* + *-Grade: int*

Grade an integer type value. It is a private attribute. Every post has a grade which is given by reviewers.

* + -isApproved

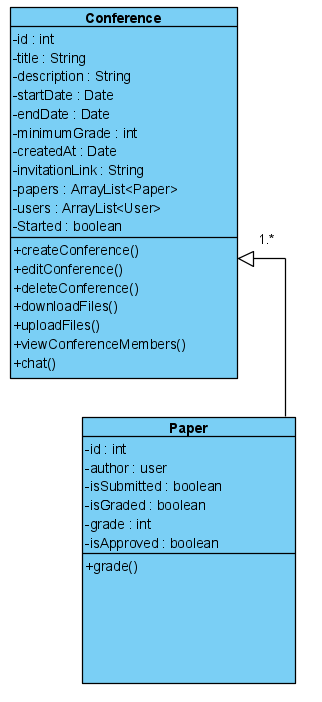
IsApproved is one of the papers information of type boolean and it is a private attribute. It provides to see paper is approved or not.

**3.1.2 Operations**

* + +Grade() :

Grade() is a void type of function, it is a public function. When this function is invoked, a paper will be graded and added to the firebase database.

**3.1.3 Dependencies with other classes and packages:**

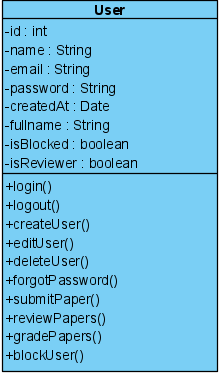


The Paper class interacts with Conference class. When a user create or join conference, if the user wants to add his/her papers, Paper class interacts with conference class

**3.1.4 Exceptions Raised and Exception Handling**

When some unexpected(not due to user error) failures occurs during adding or editing or deleting paper process, a message is displayed on the screen informing user of the failure.

**3.1 User Class**



Class Name: User

**3.2.1 Attributes**

* *-Name : String*

Name is a user identification number of type string. It is a private attribute. It is used in order to authenticate the user with the system.

* *-id : int*

Id is a user identification number of type positive integer. It is a private attribute. It is used in order to distinguish the user from other users.

* *-email : String*

Email is one of the user’s information of type string and it is a private attribute. It is used for some tasks like when user forgot the password and in order to authenticate the user with the system. He/she can get reset password link via his/her email.

* *-password : String*

Password is a user identification number of type string. It is a private attribute. It is used in order to authenticate the user with the system.

* *-fullName : String*

fullName is one of the user’s information of type string and it is a private attribute. It is showed user’s profile, on posts, comments and search pages.

* *-createdAt : Date*

createdAt is one of the user’s information of type Date and it is a private attribute. When visitor is registered and become a registered user, there is an information that stores the date the creation of user account.

* *-isBlocked : Boolean*

isBlocked is one of the user’s information of type boolean and it is a private attribute. It provides user to block other users and so banned person can’t see this user’s post and without unlocking the block, banned user can’t follow this user.

* *-isReviewer : Boolean*

isReviewer is one of the user’s information of type boolean and it is a private attribute. It provides user to see if he/she is reviewer or not.

**3.2.2 Operations**

* *+logout()*

logout() is a public function which is used for finalize the user’s session. When a user wants to exit from the system. He/she must use this function to finalize his/her session.

* *+login()*

login() is a public function which is used for authentication of user. When a user wants to enter the system, he/she must use this function. To access user related functions, first user is needed to be logged in.

* *+createUser()*

createUser() is a public function that does not return a value. When this function is called, user created with given values.

* *+forgotPassword()*

forgotPassword() is a public function which is used for adjusting new password when user forget his/her account password. It can be done by tying an email and clicking the link in the email and then user can determine his/her new password for account.

* *+deleteUser()*

deleteUser() is a public function which is used for deleting user account. When a user wants to delete his/her account, this can be done on user menu by choosing delete account and clicking okay button and then user related information would be deleted on the system.

* *+editUser()*

editUser() is a public function which is used for editing editable user information that is showed on user profile. E-mail, name and username can be changed by using edit profile function.

* *+submitPaper()*

submitPaper() is a public function which is used for submitting papers at conferences.

* *+reviewPapers()*

reviewPapers() is a public function which is used for reviewing papers for only reviewers. Only usable by reviewer users.

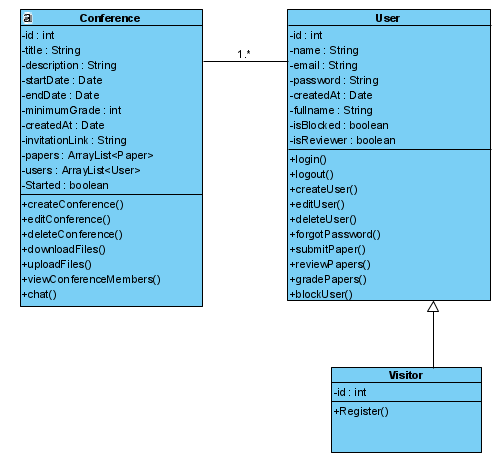
* *+gradePapers()*

gradePapers() is a public function which is used for grading papers of conferences contributors. Only usable by reviewers.

* *+blockUser()*

blockUser() is a public function which is used for blocking any user for that account so banned user can’t see the messages of user who banned himself/herself.

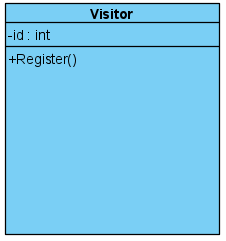
**3.2.3 Dependencies with other classes**



**3.2.4 Exceptions Raised and Exception Handling:**

When user try to edit their profile but s/he have missing part they couldn’t fill the system response “spaces cannot be empty” and also while try to login the system user write wrong information system return “wrong password or username” messages.

**3.3 Visitor Class**



Class Name:Visitor

**3.3.1 Attributes**

* -id : int

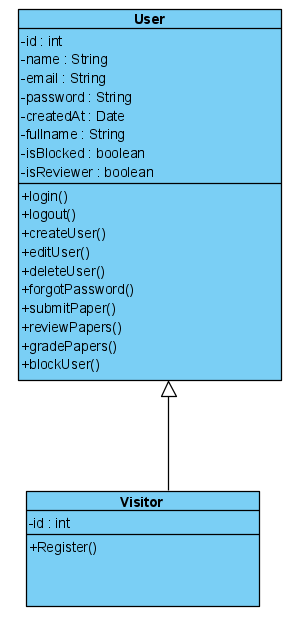
Id is a user identification number of type positive integer. It is a private attribute. It is used in order to distinguish the user from other users.

**3.3.2 Operations**

* +registerUser()

registerUser() is a public function which is used for registration of visitor type of user so when user register to the system he/she can able to use other user related and system functions.

**3.3.3 Dependencies with other classes:**

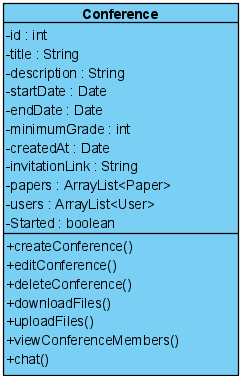


The Visitor class interacts with User class. Visitor class is inherited from user class and have only one function that this type of users can register to system. After registration that user can be able to use other user related functions such as login, submit papers, edit user, block other users etc.

**3.3.4 Exceptions Raised and Exception Handling:**

There is an exception that can be raised by the Visitor class when using register function. when user wants to register with information such as name, password, and email, some of them may raise an exception and system display an error message to user that says password or email is not valid.

**3.4 Conference Class**



Class Name: Conference

**3.4.1 Attributes**

* *-id : int*

Id is a user identification number of type positive integer. It is a private attribute. It is used in order to distinguish the user from other users.

* *-title : String*

title is one of the conference’s information of type string and it is a private attribute. Every conference should have own title .

* *-description : String*

description is one of the conference’s information of type string and it is a private attribute. Every conference can have own description.

* *-startDate : Date*

startDate is one of the user’s information of type Date and it is a private attribute. When user create a conference ,user must set start date of conference.

* *-endDate : Date*

endDate is one of the user’s information of type Date and it is a private attribute. When user create a conference ,user must set end date of conference.

* *-minimumGrade :int*

MinimumGrade is one of the conference’s information of type int and it is a private attribute. When user create a conference, user must set minimumGrade of papers which can contribute to conference.

* *-createdAt : Date*

createdAt is one of the conference’s information of type Date and it is a private attribute. When user creates conference, there is an information that stores the date the creation of conference.

* *-invitationLink: String*

description is one of the conference’s information of type string and it is a private attribute. Every conference have own invitation link. It is used in order to distinguish the conference from other conference. With this link users can join conference.

* *-papers: ArrayList<Paper>*

Papers is one of the conference’s attributes of type arraylist takes a Paper object. It is a private attribute. It keeps the user papers and these are showed on reviewer page of conference.

* *-users : ArrayList<User>*

Users is one of the conference’s attributes of type arraylist takes a user object. It is a private attribute. It keeps the users and these are showed on viewer paper of conference.

* *-isStarted: Boolean*

isStarted is one of the conference’s information of type boolean and it is a private attribute. With this information users can understand that conference is started or not.

**3.4.2 Operations**

* *+createConference()*

createConference() is a public function which is used for creating conference. User must give following information: title, description, startDate and endDate. After giving this information application creates conference succesfully.

* *+editConference()*

editConference() is a public function which is used for editing editable conference information that is showed on conference. Title, description, startDate and endDate can be changed by using edit conference function.

* *+deleteConference()*

deleteConference() is a public function which is used for deleting conference. When a user wanted to delete his/her conference, this can be done on conference menu by choosing delete conference and clicking okay button and then selected conference would be deleted on the system.

* *+downloadFiles()*

downloadFiles() is a public function which is used for downloading papers that used at conference. Only usable for reviewers.

* *+uploadFiles()*

uploadFiles() is a public function which is used for uploading papers that will use at conference. All user can use this operation.

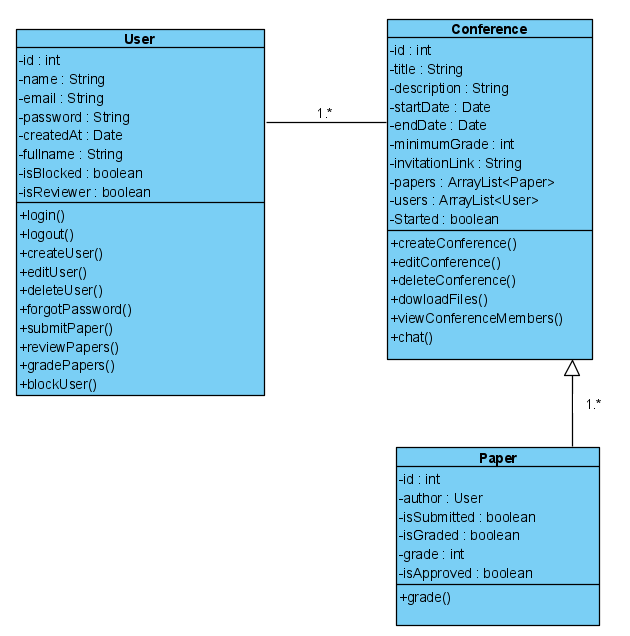
* *+viewConferenceMembers()*

ViewConferenceMembers() is a public function which is used for viewing the list of contenders of the conferences.

* *+chat()*

Chat() is a public function which is used for sending and receiving chat messages during the conference dates.

**3.4.3 Dependencies with other classes:**



The Conference class interacts with Paper and Visitor class. User is a class that visitor type of user is inherited from this class and could be created in the system. The User class can login, logout, forgot password, edit user, review papers, delete user, block user, submit paper, grade paper. By interacting with Paper class, users can do grade in conference page.

**3.4.4 Exceptions Raised and Exception Handling:**

The Conference class 3 different exception catch. First one, is about for the invitation link. If the user type wrong link or make typo error while joining conference the app should catch the error and return ‘Broken Link’ message to user.

The other one is while creating conference some datas can be empty(like start-end date etc.) this was handled by “NullPoint exception”.

Also looking for while looking for papers on the conference, if there is no papers posted the null pointer exception catch it and return no papers messages.