Jahangirnagar University

Institute of Information Technology

Easy C: A new method of learning C programming in the context of Bangladesh

Presented By:
Md. Nafis Sadique-120083
Ahmedur Rahman Shovon-120090
Md. Aslam Hossin-120091

Supervised By:
Risala Tasin Khan
Associate Professor
Institute of Information Technology
Jahangirnagar University

Outline:

- Overview
- Related Works
- Research Questions
- Objectives
- Project Diagrams
- Algorithm for Challenge Judgement
- Results
- Conclusion
- Future Work
- References
- Prototype Presentation

Overview:

- Disciplined lesson categories with required lessons.
- Category wise challenges to judge the learnt lessons.
- Ranklist to check the current position among other learners.
- Dashboard to judge the progress of learning.
- Online community with instant chatting service.
- Online C compiler with extensible syntax editor.
- Opportunity to view other learners progress.
- Ability to track own submissions.

Related Works

Online Tutorial Sites:

- No opportunity to test the learnt knowledge.
- No practice either.
- Example: Tutorialspoint, w3schools etc.

Related Works(cont’d)

Online Judges:

- No opportunity to learn the lessons.
- No fixed courseware.
- Example: Uva online judge, codeforces, codechef

Research Questions:

- How to provide online self maintained code execution environment with Sandbox support?
- How to combine challenges and lessons to create a disciplined courseware?
- How to keep track of learners progress?
- How the learners can interact with each other?

Objectives:

- **Easy C** provides detailed tutorials on C programming language in 16 predefined categories and problems related to the learnt knowledge.
- Using this application the learners can write code, execute and get verdict of the code online using any device having a web browser.
- **Easy C** keeps track of each user’s progress and give a system which can meet the teachers demand of C programming language course.
- It has both web application and Android mobile application version.

Project Diagrams

Project Diagrams:

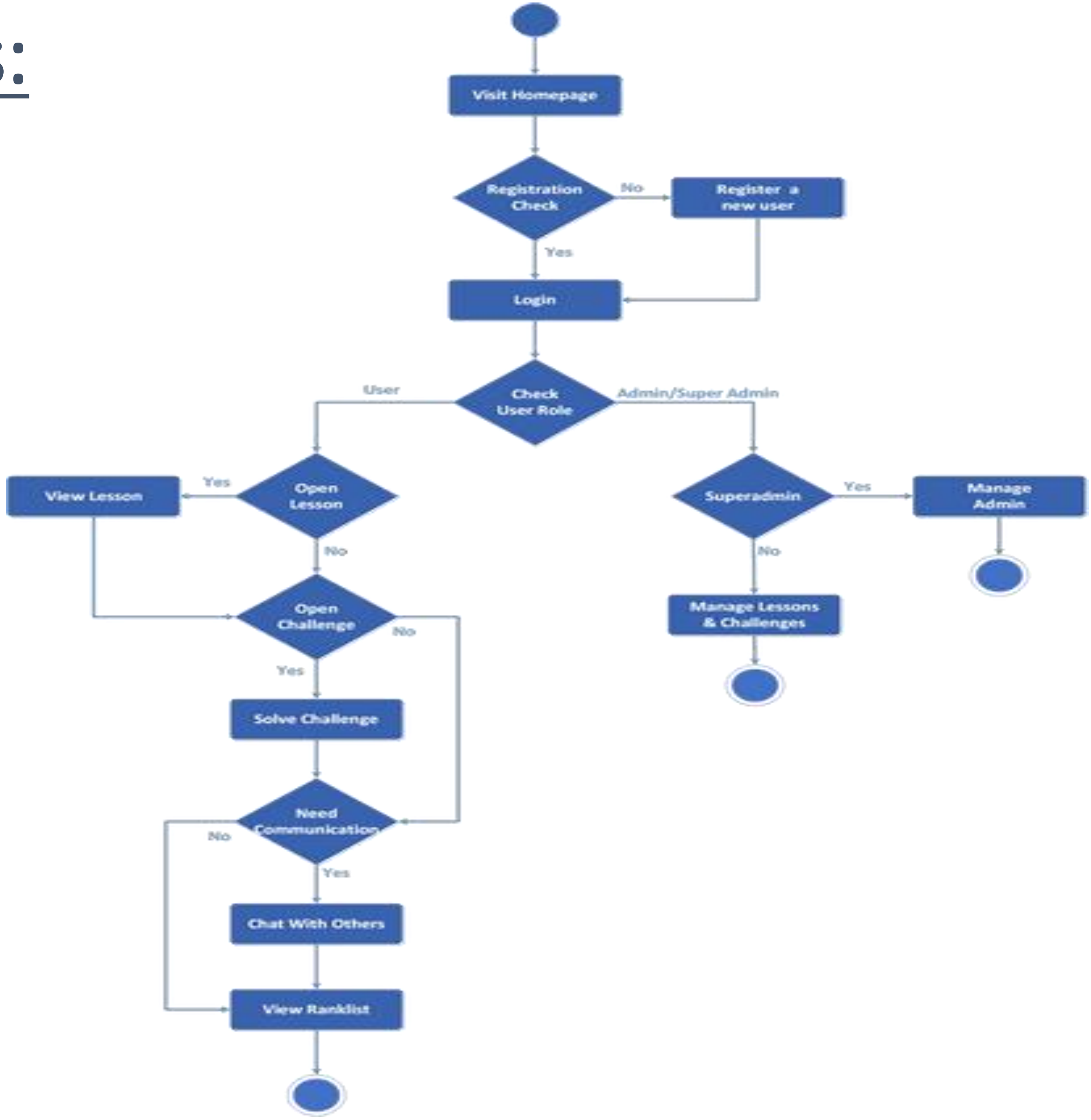


Figure :Activity Diagram

Project Diagrams(cont'd):

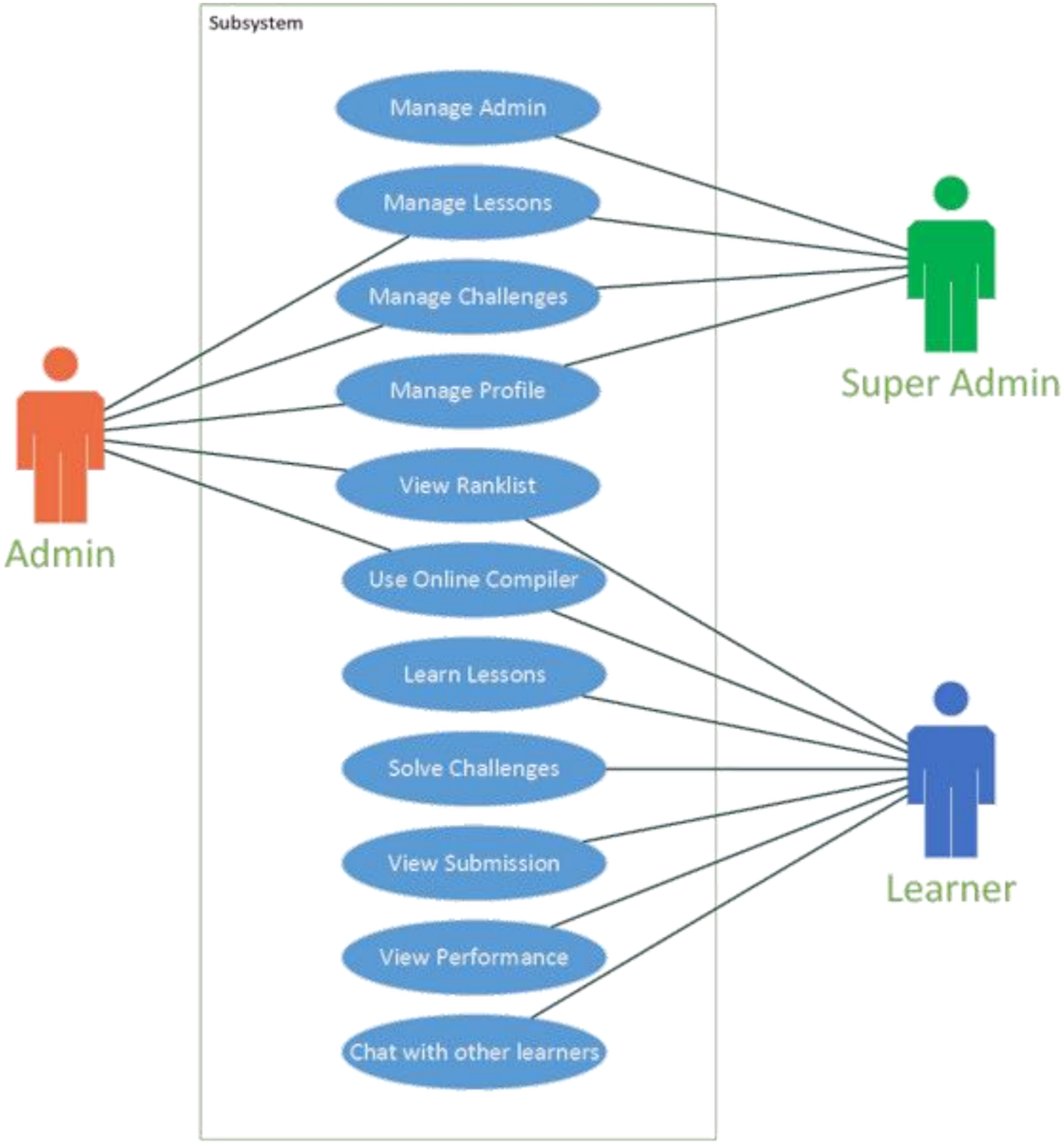
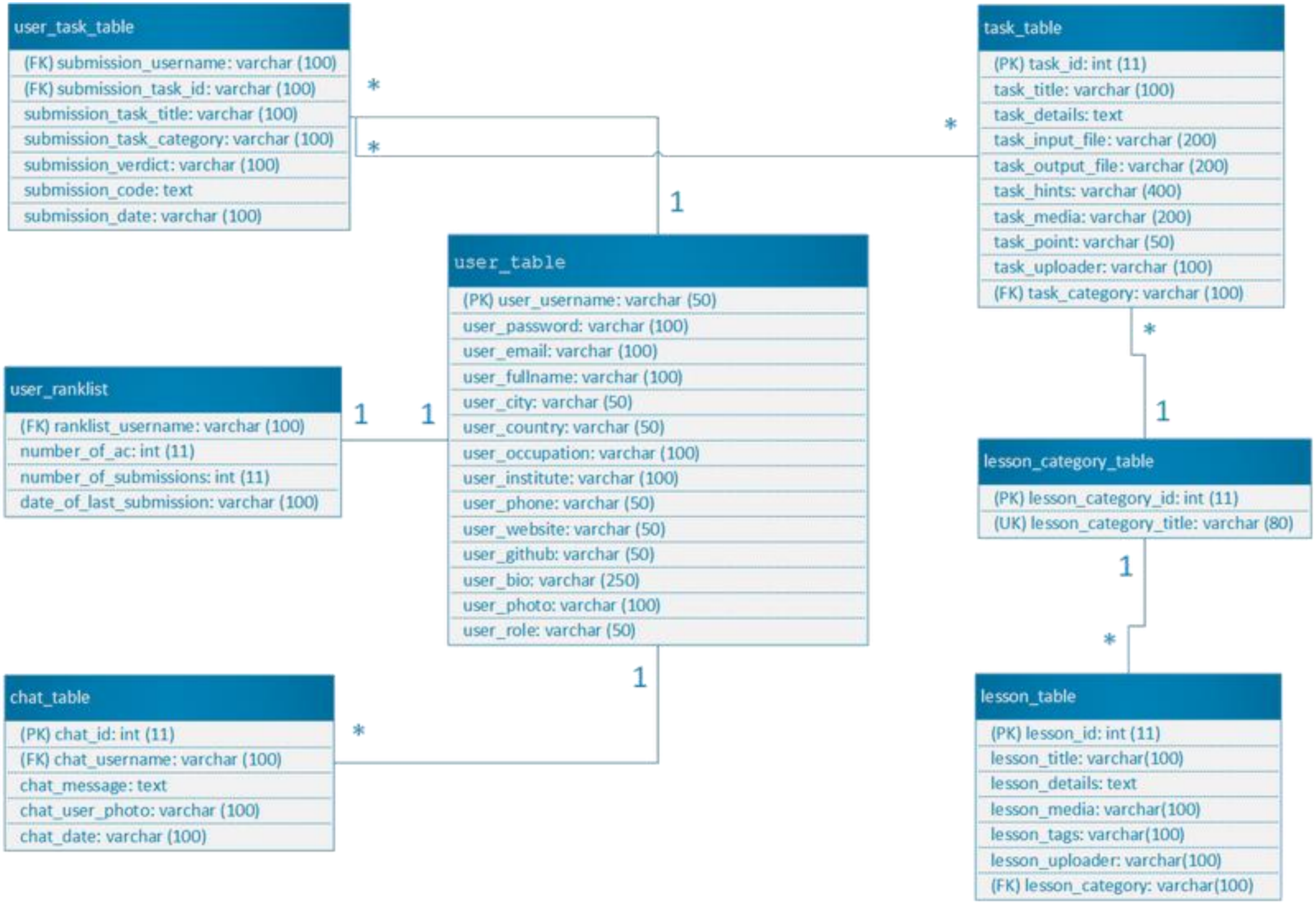


Figure: Use-case Diagram

Project Diagrams(cont’d):



user_task_table
(FK) submission_username: varchar (100)
(FK) submission_task_id: varchar (100)
submission_task_title: varchar (100)
submission_task_category: varchar (100)
submission_verdict: varchar (100)
submission_code: text
submission_date: varchar (100)

task_table
(PK) task_id: int (11)
task_title: varchar (100)
task_details: text
task_input_file: varchar (200)
task_output_file: varchar (200)
task_hints: varchar (400)
task_media: varchar (200)
task_point: varchar (50)
task_uploader: varchar (100)
(FK) task_category: varchar (100)

user_table
(PK) user_username: varchar (50)
user_password: varchar (100)
user_email: varchar (100)
user_fullname: varchar (100)
user_city: varchar (50)
user_country: varchar (50)
user_occupation: varchar (100)
user_institute: varchar (100)
user_phone: varchar (50)
user_website: varchar (50)
user_github: varchar (50)
user_bio: varchar (250)
user_photo: varchar (100)
user_role: varchar (50)

user_ranklist
(FK) ranklist_username: varchar (100)
number_of_ac: int (11)
number_of_submissions: int (11)
date_of_last_submission: varchar (100)

lesson_category_table
(PK) lesson_category_id: int (11)
(UK) lesson_category_title: varchar (80)

chat_table
(PK) chat_id: int (11)
(FK) chat_username: varchar (100)
chat_message: text
chat_user_photo: varchar (100)
chat_date: varchar (100)

lesson_table
(PK) lesson_id: int (11)
lesson_title: varchar(100)
lesson_details: text
lesson_media: varchar(100)
lesson_tags: varchar(100)
lesson_uploader: varchar(100)
(FK) lesson_category: varchar(100)

Figure: Entity Relationship Diagram

Project Diagrams(cont'd):

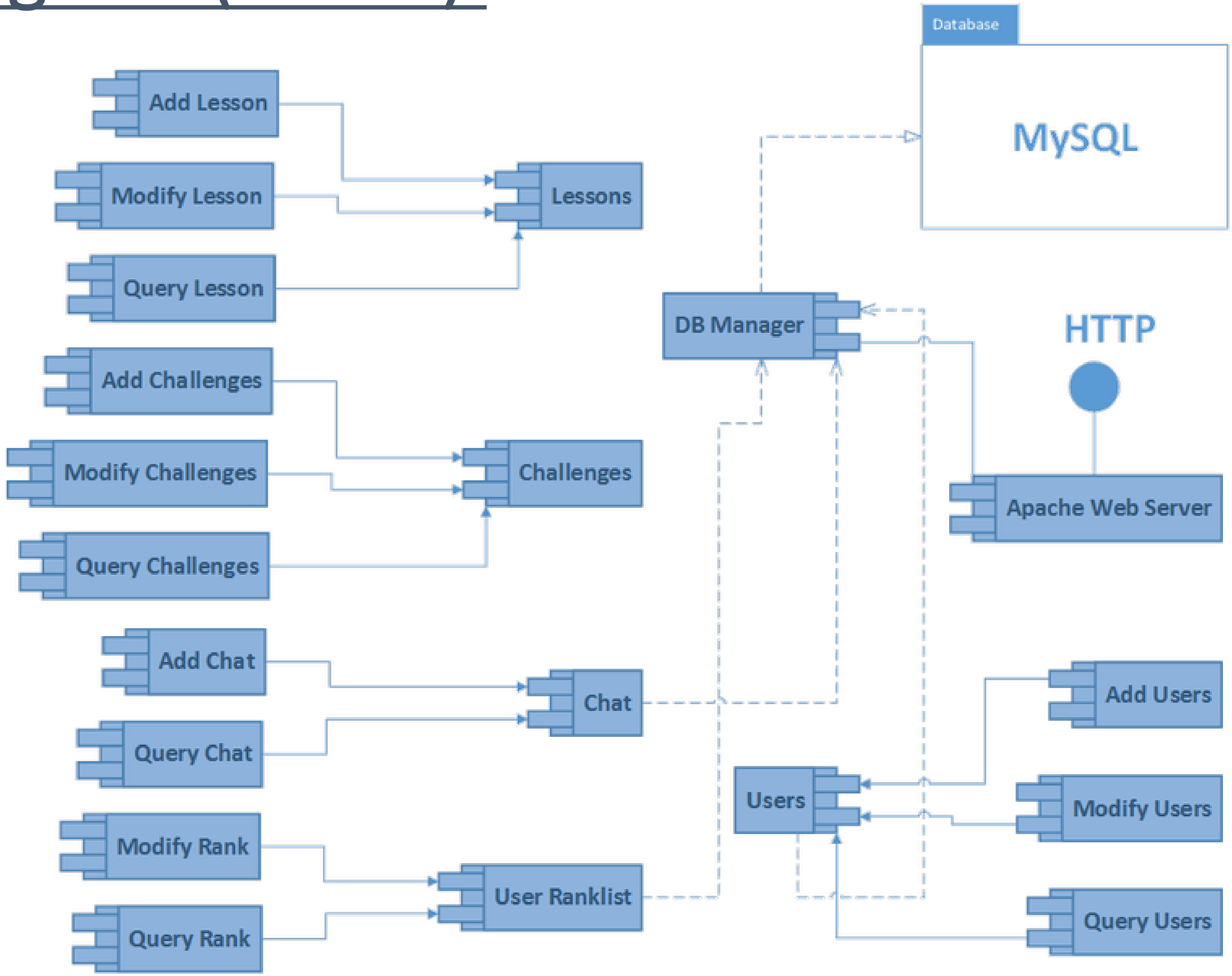


Figure: Component Diagram

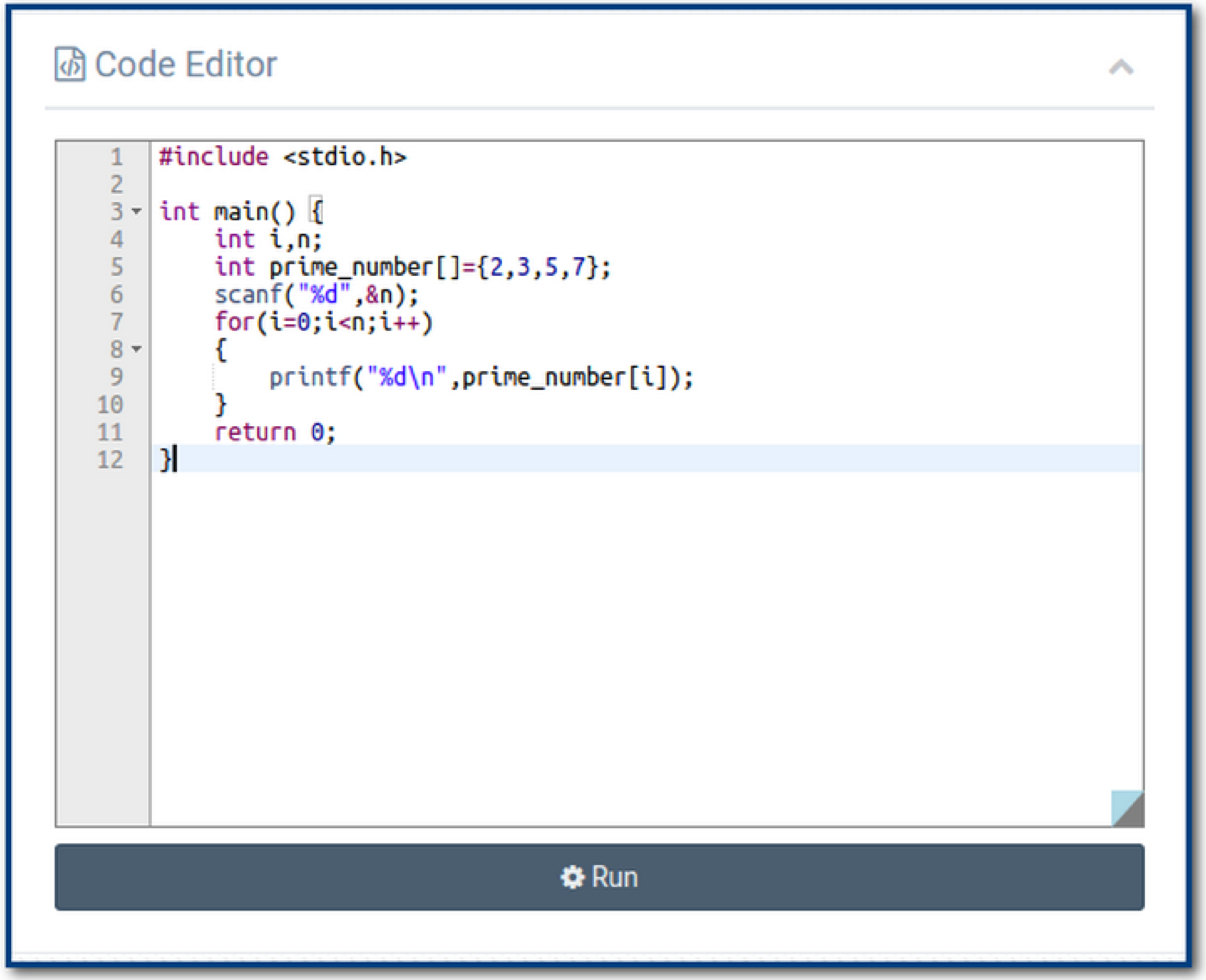
Algorithm for challenge judgement:

```
function challenge_judgement (string user_code):
    if user_code is compiled successfully
        then run user_code
        if the execution time exceeds time limit
            then return verdict Time Limit Error
        else if does not exceed time limit
            then store the output after running user_code
            if the output matches with judge output
                then return verdict Accepted
            else if does not match
                then return verdict Wrong Answer
            end if
        end if
    else if user_code does not compile successfully
        then return verdict Compilation Error
    end if
end function
```

Results

Results: Online Compiler

- **Easy C** provides full independent online compiler with Sandbox.



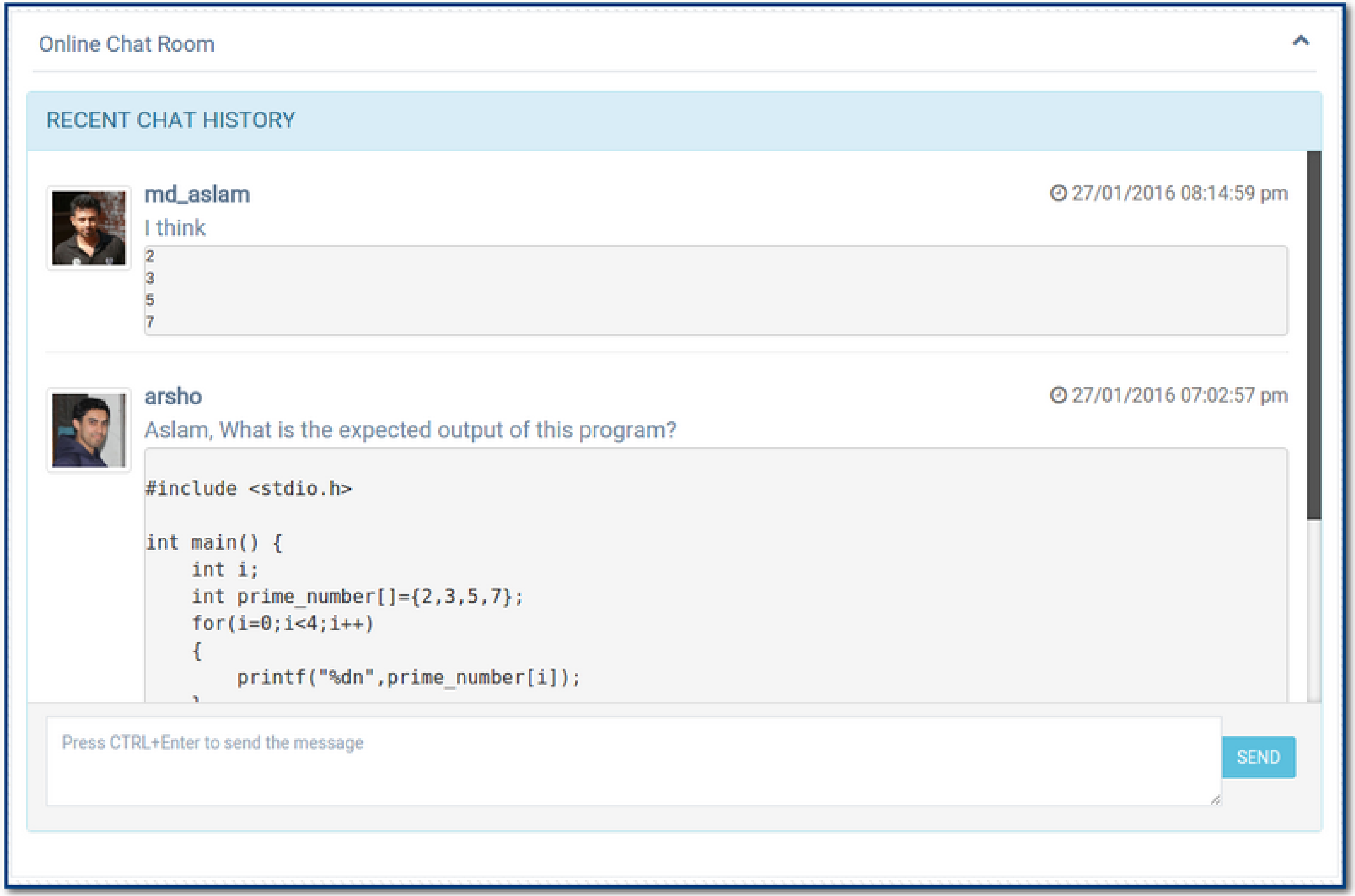
```
#include <stdio.h>

int main() {
    int i,n;
    int prime_number[]={2,3,5,7};
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("%d\n",prime_number[i]);
    }
    return 0;
}
```

Figure : Online Compiler

Results(cont’d): Instant Chatting

- Instant chatting enables to grow a community of learners.



md_aslam 27/01/2016 08:14:59 pm

I think

```
2
3
5
7
```

arsho 27/01/2016 07:02:57 pm

Aslam, What is the expected output of this program?

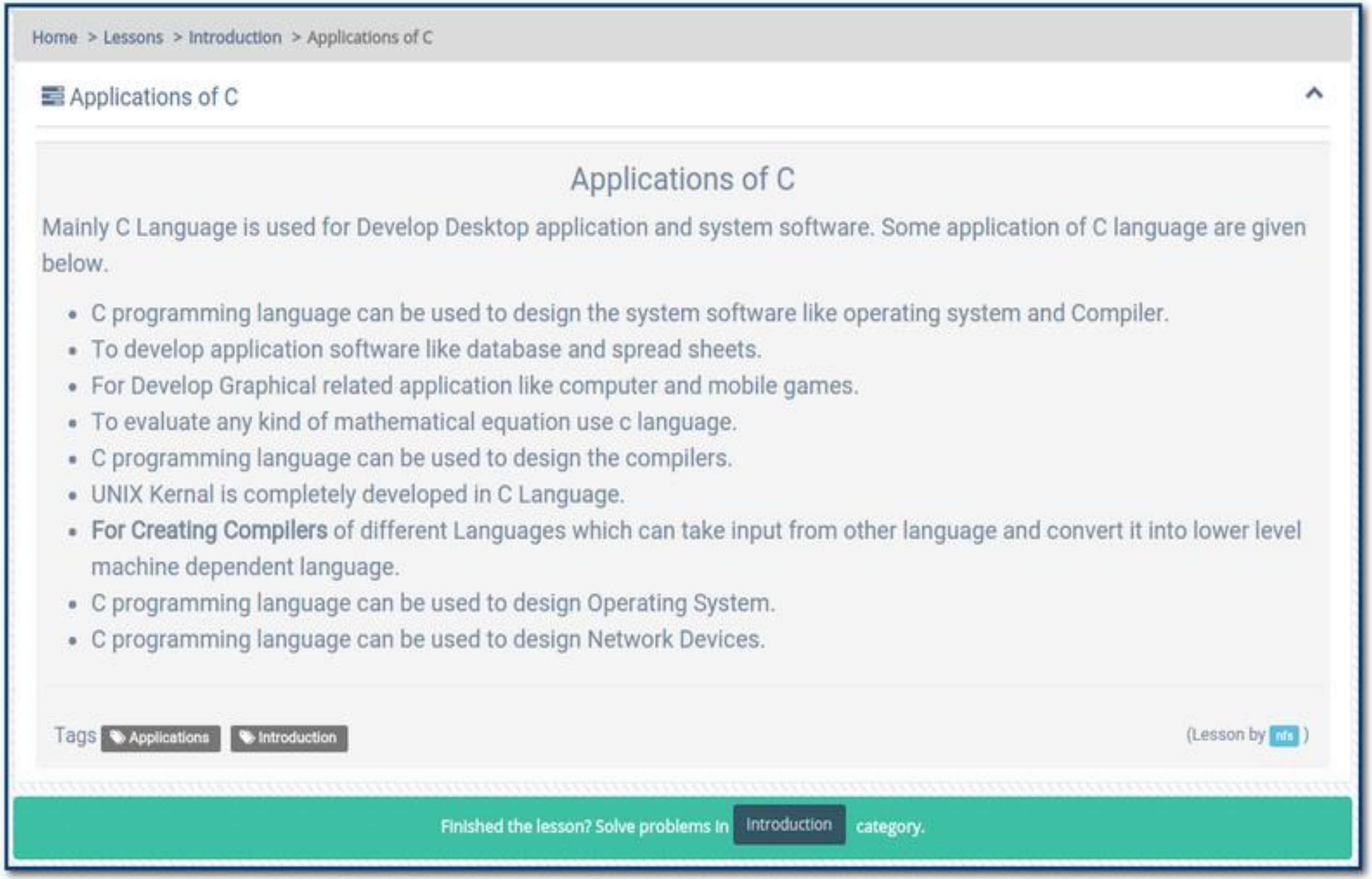
```
#include <stdio.h>

int main() {
    int i;
    int prime_number[]={2,3,5,7};
    for(i=0;i<4;i++)
    {
        printf("%dn",prime_number[i]);
```

Figure : Instant Chatting

Results(cont'd): Lessons

- Lesson of each category has singleton view.



Applications of C

Mainly C Language is used for Develop Desktop application and system software. Some application of C language are given below.

- C programming language can be used to design the system software like operating system and Compiler.
- To develop application software like database and spread sheets.
- For Develop Graphical related application like computer and mobile games.
- To evaluate any kind of mathematical equation use c language.
- C programming language can be used to design the compilers.
- UNIX Kernal is completely developed in C Language.
- **For Creating Compilers** of different Languages which can take input from other language and convert it into lower level machine dependent language.
- C programming language can be used to design Operating System.
- C programming language can be used to design Network Devices.

Tags Applications Introduction

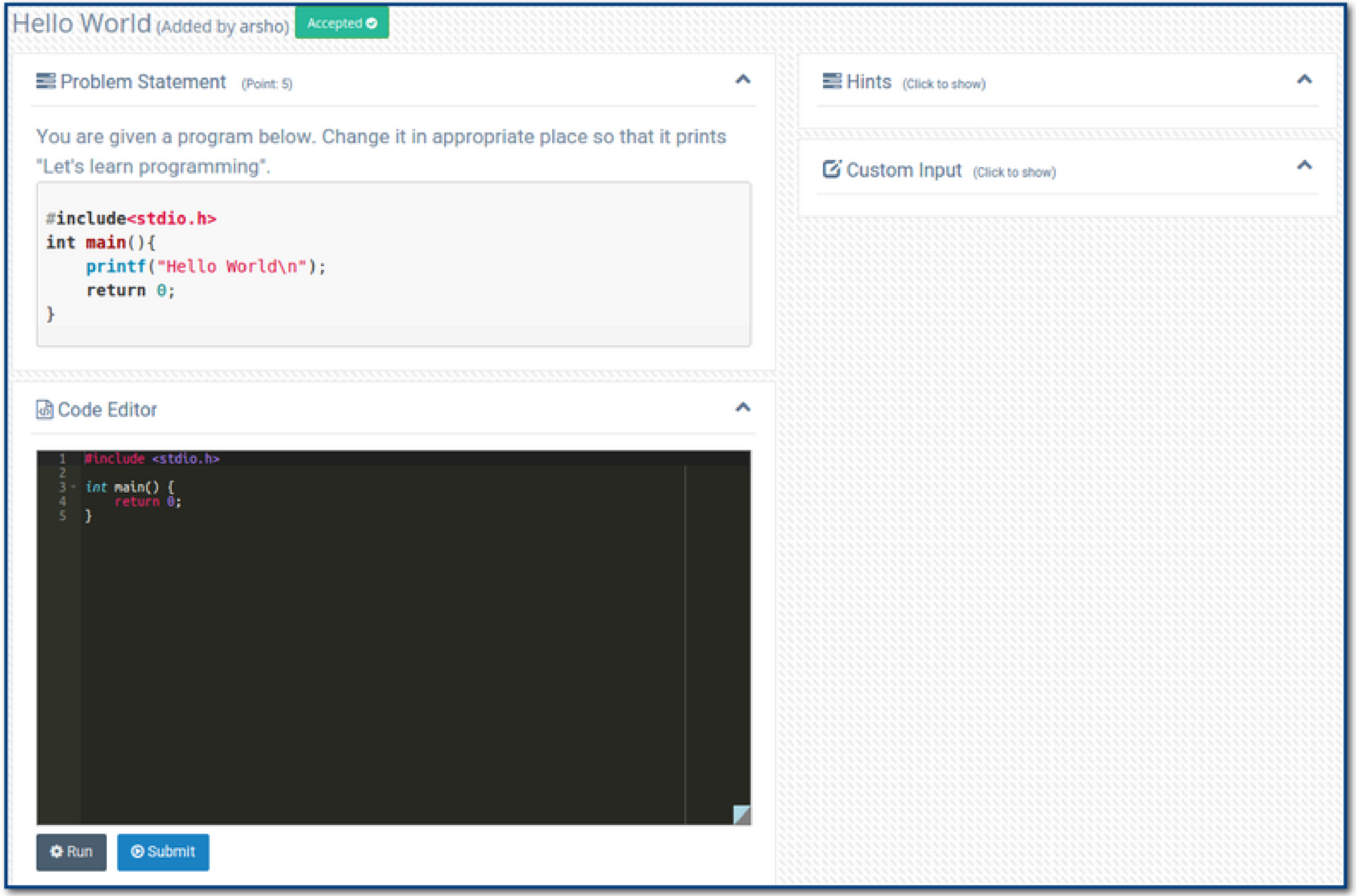
(Lesson by nfs)

Finished the lesson? Solve problems in Introduction category.

Figure : Applicatino of C in Introduction Category

Results(cont’d): Challenges

- Each category has corresponding challenges to ensure learners progress.


Hello World (Added by arsho)
Accepted
Problem Statement (Point: 5)
Hints (Click to show)
Custom Input (Click to show)
Code Editor
Run
Submit

You are given a program below. Change it in appropriate place so that it prints "Let's learn programming".

```
#include<stdio.h>
int main(){
    printf("Hello World\n");
    return 0;
}
```

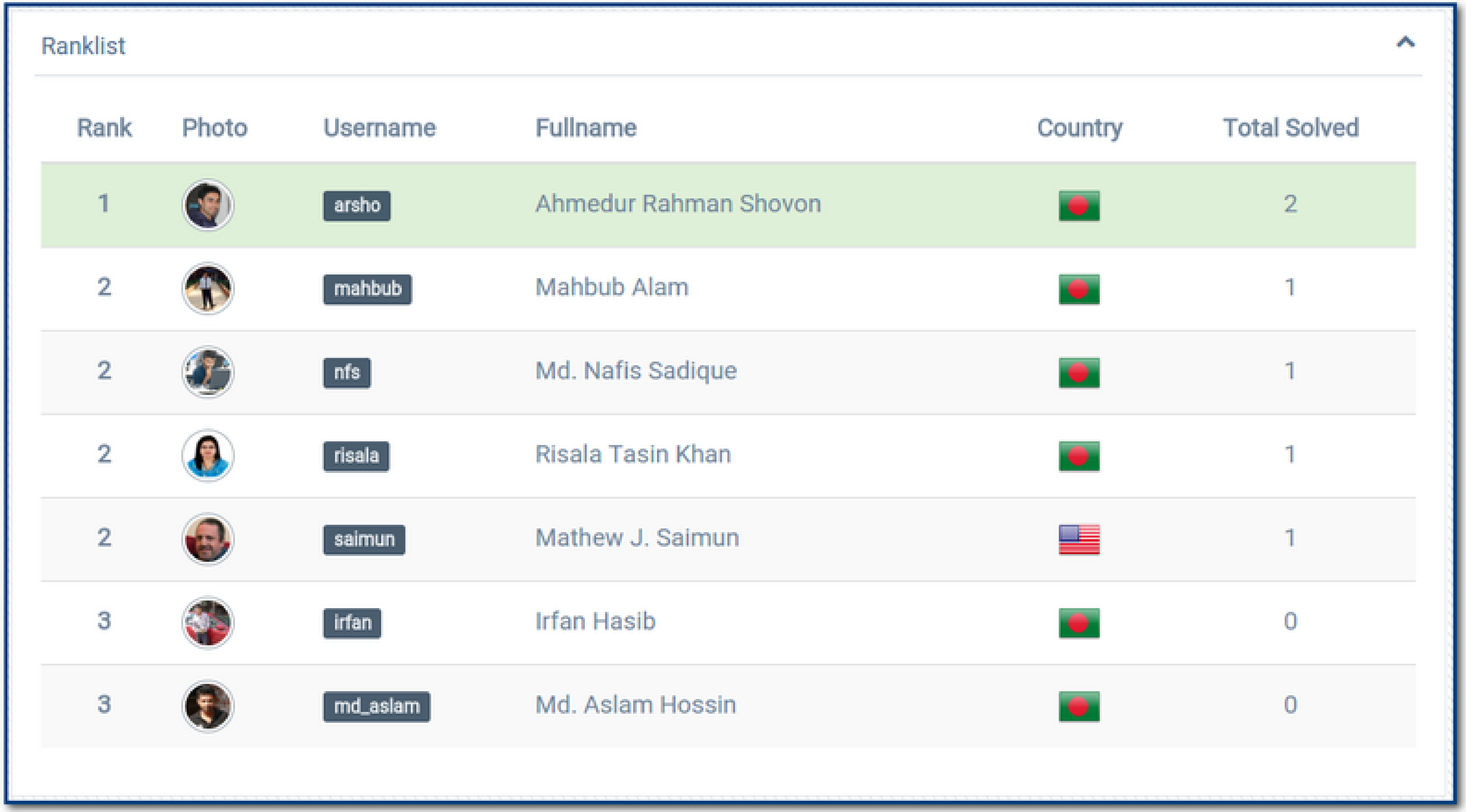
```
#include <stdio.h>

int main() {
    return 0;
}
```

Figure : Hello World Challenge in Introduction Category

Results(cont'd): Ranklist

- Learners are ranked according to the number of solved challenges.



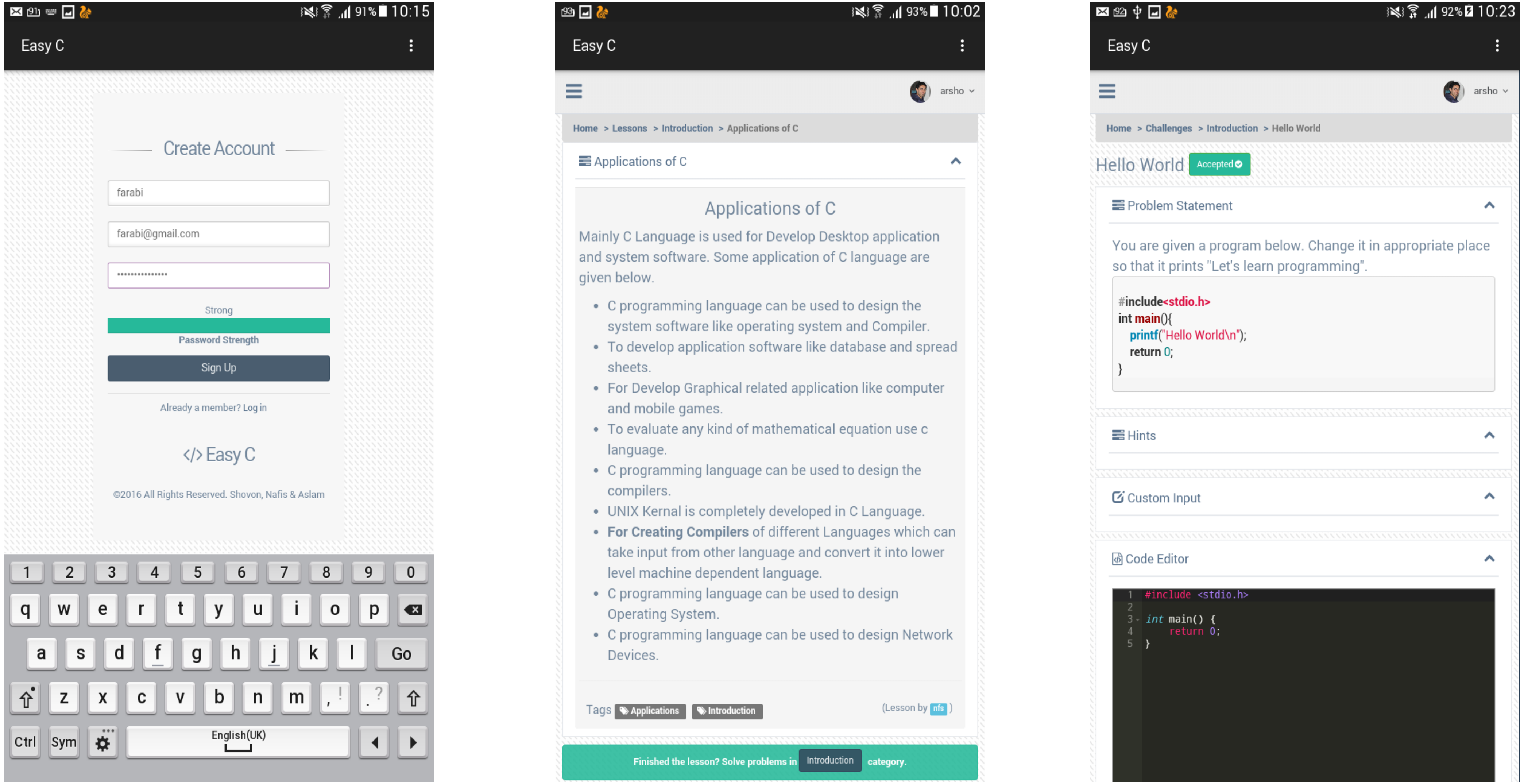
Ranklist

Rank	Photo	Username	Fullname	Country	Total Solved
1		arsho	Ahmedur Rahman Shovon	🇧🇩	2
2		mahbub	Mahbub Alam	🇧🇩	1
2		nfs	Md. Nafis Sadique	🇧🇩	1
2		risala	Risala Tasin Khan	🇧🇩	1
2		saimun	Mathew J. Saimun	🇺🇸	1
3		irfan	Irfan Hasib	🇧🇩	0
3		md_aslam	Md. Aslam Hossin	🇧🇩	0

Figure : Ranklist

Results(cont'd): Mobile Application

- Easy C:Provides Mobile application for the user



Easy C

Create Account

farabi

farabi@gmail.com

···············

Strong

Password Strength

Already a member? Log in

</> Easy C

©2016 All Rights Reserved. Shovon, Nafis & Aslam

Easy C

Home > Lessons > Introduction > Applications of C

Applications of C

Applications of C

Mainly C Language is used for Develop Desktop application and system software. Some application of C language are given below.

- C programming language can be used to design the system software like operating system and Compiler.
- To develop application software like database and spread sheets.
- For Develop Graphical related application like computer and mobile games.
- To evaluate any kind of mathematical equation use c language.
- C programming language can be used to design the compilers.
- UNIX Kernal is completely developed in C Language.
- **For Creating Compilers** of different Languages which can take input from other language and convert it into lower level machine dependent language.
- C programming language can be used to design Operating System.
- C programming language can be used to design Network Devices.

Tags Applications Introduction (Lesson by nfs)

Finished the lesson? Solve problems in Introduction category.

Easy C

Home > Challenges > Introduction > Hello World

Hello World Accepted

Problem Statement

You are given a program below. Change it in appropriate place so that it prints "Let's learn programming".

```
#include<stdio.h>
int main(){
   printf("Hello World\n");
   return 0;
}
```

Hints

Custom Input

Code Editor

```
#include <stdio.h>

int main() {
    return 0;
}
```

Figure : Easy C Mobile Application Demonstration

Conclusion:

- Lesson-Challenge paired disciplined courseware for C programming language.
- New way of interactive programming.
- Full independent online compiler with Sandbox.

Future work:

- Extending the platform for other programming languages like C++, Java, Python etc.
- Adding video tutorials.
- Arranging online contest.
- Increasing server capacity.
- Deploying “**Easy C**” in cloud server.

Reference:

- [1] Winston W. Royce: “Managing The Development Of Large Software Systems”, https://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf , April, 2007.
- [2] Youssef Bassil: “A Simulation Model for the Waterfall Software Development Life Cycle”, International Journal of Engineering & Technology (iJET), Vol. 2, November, 2012.
- [3] Kai Petersen, Claes Wohlin, and Dejan Baca : “The Waterfall Model in Large-Scale Development”,http://www.divaportal.org/smash/get/diva2:835760/FULLTEXT02.pdf, May, 2010.
- [4] Nabil Mohammed Ali Munassar, and A. Govardhan: “A Comparison Between Five Models Of Software Engineering”, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, September 2010.
- [5] Anotonia Bertolino: “Software Testing Research and Practice”, http://www.cis.upenn.edu/~lee/05cis700/papers/Ber03.pdf , October, 2011.
- [6] Laurie Williams, Gunnar Kudrjavets, and Nachiappan Nagappan: “On the Effectiveness of Unit Test Automation at Microsoft”, http://collaboration.csc.ncsu.edu/laurie/Papers/Unit_testing_cameraReady.pdf , October, 2010.
- [7] Konstantinos Xynos, Iain Sutherland, Huw Read, Emlyn Everitt, Andrew J C Blyth: “Penetration Testing and Vulnerability Assessments: A Professional Approach”, International Cyber Resilience conference, January,2005.
- [8] William G.J. Halfond, Jeremy Viegas, and Alessandro Orso: “A Classification of SQL Injection Attacks and Countermeasures”, http://www.cc.gatech.edu/~orso/papers/halfond.viegas.orso.ISSSE06.pdf, October, 2007.

Reference(Cont’d):

- [9] João-Paulo Barros, Luis Gomes: “From Activity Diagrams to Class Diagrams”, http://www.disi.unige.it/person/ReggioG/UMLWORKSHOP/Barros.pdf , August, 2011.
- [10] A.K. Bhattacharjee, R.K. Shyamasundar: “Activity Diagrams : A Formal Framework to Model Business Processes and Code Generation”, Journal Of Object Technology, Vol. 8, No. 1, January–February 2009.
- [11] PETER PIN-SHAN CHEN: “The Entity-Relationship Model Toward a Unified View of Data”, http://www.inf.unibz.it/~nutt/IDBs1011/IDBPapers/chen-ER-TODS-76.pdf, August, 2009.
- [12] Hassan Gomaa, Erika Mir Olimpiew: “The Role of Use Cases in Requirements and Analysis Modeling“, http://www.ie.inf.uc3m.es/wuscam-05/5-WUsCaM.pdf, September, 2000.
- [13] Chris Lüer, David S. Rosenblum: “UML Component Diagrams and Software Architecture - Experiences from the WREN Project”, Describing Software Architecture with UML, Workshop at the 23rd International Conference on Software Engineering, Toronto, Canada, 2001.
- [14] Antonio Carzaniga, Alfonso Fuggetta, Richard S. Hall, Dennis Heimbigner, Andr´e van der Hoek, Alexander L. Wolf: “A Characterization Framework for Software Deployment Technologies”, University of Colorado, Department of Computer Science, Technical Report: CU-CS-857-98, April 1998.

Prototype presentation