

Development of Intelligent Tutoring System Framework For Game-based Learning

A Thesis submitted
in partial fulfillment of the
requirements for the Degree of
Dual Degree (BTech + MTech)

by

Praveen Kumar

(07D05019)

under the guidance of

Prof. Sridhar Iyer



Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay
June 2012

Abstract

A lot of research and development is happening in the past few years on the use of computers as teaching tools. While many tutoring systems already exist, building an effective Intelligent Tutoring System is a challenge.

We have built a framework for development of Intelligent Tutoring Systems (ITS) that can support multiple teaching strategies. In this report, I have discussed ITS framework for a strategy called Game-based learning. Game-based learning is teaching-learning strategy composed of software applications or products that use games for learning or educational purposes. In an educational game, the instructional content is blurred with game characteristics.

We have implemented the system using PHP and MYSQL as the database model. Also an Android app is built to bring the learning process into the hands of a student.

The other 3 strategies: Scaffolding, Socratic Questioning and Guided Discovery, were implemented by my fellow-students - Chandrapal Singh, Vikash Kumar and M.Rajashekhar - respectively. In the last section we have discussed how we have integrated these 4 different teaching strategies into one system.

Keywords: Intelligent Tutoring Systems, Game-based Learning, Web Tutors.

Acknowledgements

I wish to extend my most sincere thanks to my advisor **Prof. Sridhar Iyer** for providing me with the opportunity of being a part of this project and for his consistent directions. The feedback provided by him has brought more clarity and focus in my work. He has been remarkable in his attempt to keep me motivated in this project. I would also like to thank my project team members Chandrapal, Vikash and Rajashekhar for their help in completing this project.

Finally, I wish to thank the Department of Computer Science and Engineering for providing me with the infrastructure needed.

Contents

Abstract	ii
Acknowledgements	iii
1 Introduction	1
1.1 Intelligent Tutoring System	1
1.1.1 Personalized Tutoring	1
1.1.2 Meeting Student and Instructor Goals	2
1.2 ITS Strategies	2
1.3 Organization of the report	3
2 Literature Survey	4
2.1 Examples of ITS in Education	4
2.2 Examples of ITS in Corporate Training	5
2.3 Components of Intelligent Tutoring Systems	5
2.4 Strategies	6
2.4.1 Socratic Questioning	6
2.4.2 Scaffolding	6
2.4.3 Guided discovery	6
2.4.4 Game-based learning	7
3 Game-based Learning	8
3.1 Introduction	8
3.2 Definition of GBL	8
3.3 Characteristics of Games	9
3.4 Incidental Learning	10
3.5 Typology of Games	11
3.6 Examples	11
4 ITS Support for GBL	13
4.1 Crossword	13
4.1.1 Existing Softwares	14
4.1.2 Our Model	14
4.1.3 Instructor's Role	15
4.1.4 Student's Role	15

4.1.5	Database Tables	15
4.2	Scavenger Hunt	16
4.2.1	Existing Hunts	16
4.2.2	Instructor’s Role	17
4.2.3	Student’s Role	17
4.2.4	Database Tables	17
4.3	Steps to be followed by Instructor	18
4.4	Steps to be followed by Student	19
4.5	Steps to be followed by ITS	19
4.6	Crossword Examples	20
4.7	Scavenger Hunt Example	24
5	ITS Framework and Workflow	28
5.1	Components of Intelligent Tutoring Systems	28
5.1.1	Domain Module	28
5.1.2	Student Module	29
5.1.3	Pedagogical Module	29
5.1.4	Interface Module	29
5.2	Architecture	30
5.3	Time Sequence Diagrams	31
5.3.1	Time Sequence Diagram For Instructor	31
5.3.2	Time Sequence Diagram For Student	31
5.4	Modules	32
5.4.1	GUI Module	32
5.4.2	Input Validation module	33
5.4.3	Registration Module	33
5.4.4	Authentication module	33
5.4.5	Course Module	33
5.4.6	Topic Module	33
5.4.7	Game Module	33
5.4.8	Database Handling Module	34
5.4.9	Log module	34
5.4.10	Feedback module	34
6	Android App	35
6.1	Why there is a need for mobile app?	35
6.2	Current Apps for Education	35
6.3	Activities in Android	36
6.4	Layouts	36
6.5	PHP API classes	37
6.6	JSON Responses	39
6.6.1	Types of API JSON Responses	39

7 Conclusion and Future Work	40
7.1 User Interface	40
7.2 Evaluation	40
7.3 Expanding the Scope	40
Bibliography	42

List of Tables

3.1	Topology of Games	11
4.1	Crossword table	15
4.2	Questions Table	15
4.3	Student Table	16
4.4	Scavenge Hunt table	18
4.5	Questions Table	18
4.6	Student Level Table	18
4.7	Attempted Questions Table	18

List of Figures

3.1	Learning Process in GBL [12]	9
4.1	Add Quiz	20
4.2	Add Quiz 2	21
4.3	Crossword puzzle	22
4.4	Student interface	23
4.5	Start Hunt	24
4.6	Add Questions	25
4.7	List Questions	26
4.8	Student interface	27
5.1	Components of ITS	28
5.2	Architecture of our ITS	30
5.3	Time sequence Diagram for Instructor in Our ITS	31
5.4	Time sequence Diagram for learner in Our ITS	32
6.1	PHP API Working [17]	38

Chapter 1

Introduction

Computers have become an integral part and personality of the present generation and they like to embrace all the things in the scope of this modern technology. This modern generation is much more advanced and smarter than their teachers in terms of usage of modern technical gadgets. They want the computers to be there for all their tasks including academic related activities. Keeping in mind the psychology and interest of the modern learner, many universities and educational institutions have focused on integrating the modern technology with education. In fact, a lot of research is happening on the use of computers as teaching tools.

The goal of this project is to build an Intelligent Tutoring System (ITS) framework which can guide students and help them to learn quickly and intelligently. The ITS framework built monitors the behavior of each learner and then adapts itself to the learner level. More importance is given to the user experience to make it highly interactive.

1.1 Intelligent Tutoring System

An Intelligent Tutoring System (ITS) is any computer system that provides direct customized instruction or feedback to learners, i.e. without the intervention of human beings, while performing a task. Thus, ITS implements the theory of learning by doing.

ITSs are used in many domains such as in traditional education, distance learning and training. It is a multidisciplinary area.

1.1.1 Personalized Tutoring

Research has shown that better learning outcomes result from one-on-one tutoring. It presents different content for different students depending on their individual needs

and preferences. But one-on-one interaction is not possible due to very large student-to-teacher ratio. Also, the cost and time investments required for one-on-one tutoring in any learning environment makes it very difficult to implement in today's realities. This is the reason why we need to use computers as teaching tools.

There are numerous benefits of using electronic tutors in education. These tools are called as Computer-Aided Instruction (CAI) systems. A student can learn at his own pace at anytime of the day. They provide hands-on learning on an individual basis to each student. They can be more effective than traditional text books because they can provide a higher level of interactivity through the use of visual and audio simulation. The main drawback with these systems is they are static systems and do not adapt itself according to the intelligence of the learners. This is why the need for Intelligent Tutoring Systems arises. ITS has the ability to adapt itself depending on the domain and the cognitive profile of a student. Intelligent tutoring systems can personalize learning for a diverse generation of students with changing needs. It provides a more convenient, less expensive solution by keeping the costs down and more effectively using instructor's time.

1.1.2 Meeting Student and Instructor Goals

An Intelligent Tutoring System can also benefit instructors to manage learning objectives. For example, WebStudy or Angel Learning has an integrated course management system which allows an instructor to view the progress of students learning objectives for the entire course through a dashboard reporting feature. These type of electronic tutoring systems can help instructors determine what material is presented to each student and create rules to release additional materials to students based on their assessment results. Ideally, a student and an instructor can see immediately what progress the student is making towards a learning objective. It is exciting to think that presentation of course content can adapt in response to student performance and be driven by the combined goals of student and instructor.

1.2 ITS Strategies

The ITS framework built includes the following four teaching strategies:

1. Socratic Questioning
2. Scaffolding
3. Guided Discovery and
4. Game-based Learning

My contribution to ITS Framework project is implementation of Game-based Learning.

1.3 Organization of the report

Chapter 2 contains the literature survey done on various ITSs given by different authors. It also describes our definition of ITS and its scope.

Chapter 3 describes the details of ITS based on Game-based Learning. It contains the various definitions of Game-based learning, how learning is achieved through games, and also about the characteristics of games. Various topologies of games is also explained in this chapter. Some of the examples where Game-based Learning is used as a pedagogical approach is also given.

Chapter 4 contains the list of games used in our ITS for Game-based Learning. Each game is explained in detail how it is built and the learning outcomes at the end of it. The steps that should be followed by the instructor, student and ITS is discussed in this chapter.

Chapter 5 describes the components and architecture of the ITS framework for Game-based Learning. It also contains information about modules, databases and the screenshots of some examples. The functionality and use of each module is explained clearly.

Chapter 6 describes about the android app built for our ITS. Why there is a need for android app is mentioned and also the components of the app are described clearly.

Chapter 7 discusses the conclusion and the future work.

Chapter 2

Literature Survey

Intelligent Tutor Systems have been around since the late 1970s, but increased in popularity in the 1990s. Intelligent tutoring has had relatively little impact on education and training in the world with the exception of few success stories. One of the reasons for this lack of penetration is the cost to develop Intelligent tutoring systems. Until relatively recently, the necessary computing power was expensive to deploy. But the main reason is that work on intelligent computer tutors is done mainly by Artificial Intelligence researchers rather than educational specialists. Researchers mainly used intelligent tutoring systems to deploy and improve AI algorithms. These intelligent tutoring systems are used to study and interpret student behaviors rather than analysing the educational effectiveness[3]. For intelligent tutors to seriously penetrate the educational/training system, the evaluative focus must begin to shift to educational impact and away from artificial intelligence sufficiency.

The first intelligent tutor SCHOLAR was developed in 1970 to teach South American geography using mixed-initiative dialogue. This tutor was constructed around a semantic network model of domain knowledge. The system and student communicated through a sequence of natural language questions and answers.

2.1 Examples of ITS in Education

The Mathematics Tutor helps students solve word problems using fractions, decimals and percentages. The tutor records the success rates when student is working on the problems and subsequent problems that are predicted to fit student's level will be selected and an estimated desirable time will be given to the student to solve the problem.

Why2-Atlas is an ITS that analyses students explanations of physics principles. The students input their work in paragraph form and the program converts them into a proof by making assumptions of student beliefs based on their explanations. In doing, misconceptions and incomplete explanations are highlighted. The system then addresses these issues through a dialogue with the student and asks the student

to correct their essay. A number of iterations may take place before the process is complete.

REALP was designed to help students enhance reading comprehension by providing reader-specific lexical practice and offering personalized practice in useful, authentic reading materials gathered from the Web. The system automatically builds a user model according to student's performance. After reading, the student will be given a series of exercises based on the target vocabulary found in reading.

eTeacher is an intelligent agent that supports personalized e-learning assistance. It builds student's profile while observing student performing in online courses. Then eTeacher uses the information from student's profile to suggest their personalized courses of action that assist their learning process.

2.2 Examples of ITS in Corporate Training

SHERLOCK is used to train Air Force technicians to diagnose problems in the electrical systems of F-15 jets. The ITS creates faulty schematic diagrams of systems for the trainee to locate and diagnose. The ITS provides diagnostic readings allowing the trainee to decide whether the fault lies in the circuit being tested or if it lies elsewhere in the system. Feedback and guidance are provided by the system and help is available if requested.

Cardiac Tutor: The Cardiac Tutor's aim is to support advanced cardiac support techniques to medical personnel. The tutor presents cardiac problems and, using a variety of steps, students must select various interventions. Cardiac Tutor provides clues, verbal advice, and feedback in order to personalize and optimize the learning. Each simulation results in a detailed report on student's performance.

2.3 Components of Intelligent Tutoring Systems

One of the first architectures of an ITS system was presented by Burn and Caps in 1988[1]. This architecture was based on four main components: curriculum module, student module, tutor (pedagogical module) and the interface module between the student and the system. The architecture used in our ITS is very similar to the Burn and Caps proposed architecture. The four modules are represented as the domain module, student module, pedagogical module and the interface module.

2.4 Strategies

One of the advantages provided by our Intelligent Tutoring Systems (ITS) is it allows a flexible pedagogical approach according to the cognitive ability of a learner. This flexibility is enhanced using multiple learning strategies that can be successively triggered depending on the progression of learning. The following four teaching strategies are used to build our ITS.

2.4.1 Socratic Questioning

Socratic Questioning is an approach in which teaching-learning is performed in the form of question and answer. It is a kind of series of questioning in which an original question is splitted into more than one low level questions. It is just like bottom up approach. In this strategy we start from the question which student or learner knows and goes to our target questions which we want to teach him.

Socratic Questioning is basically a dialogue conversion between the teacher and student. First, instructor starts the question and student responses. In return, instructor reformulates a new question according to the response given by student. Questioning and answering is structured systematically to reach an ultimate goal.

This strategy was implemented by Vikash Kumar [6].

2.4.2 Scaffolding

Scaffolding also known as Instructional scaffolding is a learning process designed to promote a deeper learning. Scaffolding is the support given during the learning process which is tailored to the needs of the student with the intention of helping the student achieve his/her learning goals. The best and most effective use of instructional scaffolding helps the learner figure out the task at hand on their own. Once students build knowledge and develop skills on their own, elements of the support are removed.

According to McKenzie (1999), the defining features of successful scaffolding include clear direction, purpose, and expectation. Results include on-task activity; better student direction; reduced uncertainty, surprise, and disappointment; increased efficiency; and palpable momentum.

This strategy was implemented by Chandrapal Singh [7].

2.4.3 Guided discovery

Guided discovery, also known as an inductive approach, is a technique where a teacher provides examples of a language item and helps the learners to find the rules themselves. Guided Discovery, is characterized by convergent thinking. The instructor devises a series of statements or questions that guide the learner, step by

step, making a series of discoveries that leads to a single predetermined goal. In other words the instructor initiates a stimulus and the learner reacts by engaging in active inquiry thereby discovering the appropriate response.

Guided discovery is regarded by many teachers as an important tool. It encourages independence, makes learning more memorable, and if analysis is done in groups is a meaningful communicative task.

This strategy was implemented by M. Rajashekar [8].

2.4.4 Game-based learning

Game based learning (GBL) is a branch of serious games that deals with applications that have defined learning outcomes. Generally they are designed in order to balance the subject matter with the gameplay and the ability of the player to retain and apply said subject matter to the real world.

GBL uses competitive exercises, either pitting the students against each other or getting them to challenge themselves in order to motivate them to learn better. Games often have a fantasy element that engages players in a learning activity through a storyline.

A detailed description about Game-based learning is provided in the next chapter.

Chapter 3

Game-based Learning

3.1 Introduction

The main challenge faced by the current educational system is that the students are not always eager to learn or do difficult things. Unfortunately, the content that should be learned by them is not motivating them directly and many students consider it as "boring", "dry" and "technical". It is probably safe to say that today's teachers, trainers, and educators are rarely as effective as they might be in the motivational department, and this often causes real problems in getting the otherwise highly-stimulated students to learn.

The attitude and interest of today's students towards their computer games is the very opposite of the attitude that most of them have towards school. Yet it is the very attitude that students should have towards learning: interested, competitive, results-oriented, actively seeking information and solutions.

It therefore makes a great deal of sense to try to merge the content of learning and the motivation of games, and this is what we are doing in this project. The game based learning model is used in formal education very successfully, in particular, in military, medicine, physical, etc. training.

3.2 Definition of GBL

Game-based learning (GBL) refers to different kinds of software applications or products that use games for learning or educational purposes. These game applications are also termed as serious games. They can be anything from simpler games to games which use 3D graphics to provide opportunity for learners to take on virtual presence in virtual world.

"The underlying idea is that students learn better when they are having fun and are engaged in the learning process" [10].

The main characteristic of an educational game is the fact that instructional content is blurred with game characteristics. The game should be motivating, so the learner repeats cycles within a game context. While repeating the act of playing the game, the learner acquires the knowledge through thought, experience, and the senses which result from interaction with and feedback from the game play.

Garris et al.[11] suggested that the learning outcomes occur outside of the game during reflection and debriefing. The following figure describes the debriefing process between the game cycle and the achievement of the learning outcomes.

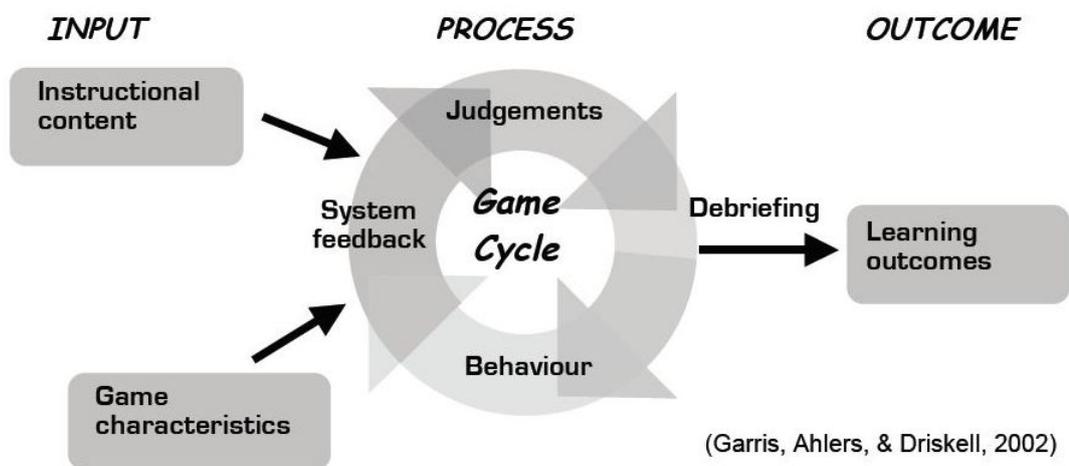


Figure 3.1: Learning Process in GBL [12]

The debriefing process provides a link between simulation and the real world. It connects game experience and learning by drawing a relationship between the game events and the real-world events. This process of study in a game is described by Kolb et al., as "doing, reflecting, understanding, and applying".

3.3 Characteristics of Games

Many authors gave different opinions about what the game characteristics are:

- Interactivity is an essential aspect of a game [Thornton et al., 90]
- The dynamic visuals, rules, goal and interaction are the essential features [Johnston et al., 93]
- The essence of playing is challenge and risk [Baranauskas et al., 99]

Malone (1981) summarized four essential characteristics or elements for computer games to answer the question of what makes a computer application enjoyable to operate: fantasy, curiosity, challenge and control.

Fantasy stands for the virtual world or the scenario in which the activity is embedded. Games involve imaginary worlds and nothing outside the game is relevant. The activity inside the game has no impact on the real world. The fantasy in the context of the game leads to greater interest on the part of the student as well as increased efficiency of learning.

Curiosity is embedded into the game by the continual introduction of new information and non-deterministic outcomes. The different types of rules help the players to reach the goal of the game. This includes the use of randomizing to add variety to the game. Malone emphasizes that environments should be neither too complicated nor too simple with respect to the end-user's existing knowledge.

Challenge is provided within each appropriate level of difficulty. If the activity level of difficulty is too low, students lose interest and the same occurs if the activity level is too high relative to the students abilities. The challenge should be incorporated by introducing multiple levels or goals with progressively increasing the difficulty levels

The students **control** the game environment by making the decisions and choices. This opportunity to make choices has direct consequences and helps to solve the confronted tasks and problems.

3.4 Incidental Learning

Incidental learning is characterized by discovering something while in the process of doing something else. In the traditional method of classroom learning, intentional learning dominates. But with the aid of a game, incidental learning is possible because the primary intent of the players is to accomplish the game rather than to learn.

According to Lankard (1995), "incidental learning occurs when it is unexpected - a byproduct of other activities, such as, for example, playing a game". Although students play the game on a voluntary basis, their primary intent was of course not to learn. After playing the game once, a learning process occurs.

3.5 Typology of Games

Learning Objective	Features	Game Type	Examples
Memory; Retention	Content Engine; Assessment Engine	Drill and Practice; Quiz; Puzzle	MedicMED; Children Software
Precision; Motoric	Game engine; time factor; sessions	Shoot'em up; Driving; Simulation	DOOM; Flight simulator; Tomb Raider
Applying Concepts and Rules	Rules and instructions; Reality vs abstraction	Sports, action, driving games	FIFA 2004; Monkey Wrench Conspiracy
Decision making; Problem Solving	Narrative; Real Time monitoring; open endedness	Strategic Games; Adventure games; Role Play	SIM saga; Business game builder
Social Integration; Values; Cultures;	Communication modules; Narrative; Tracking	Strategic games; Role Play games	Collective bargaining; Unigame

Table 3.1: Topology of Games

3.6 Examples

There are several projects and initiatives which used Game-based Learning as a pedagogical approach. Some of them are:

Environmental Detectives (2002) was developed by MIT (Massachusetts Institute of Technology) and Microsoft within the Games-to-Teach project where conceptual prototypes for the next generation of interactive educational entertainment are developed. Environmental Detectives is designed to be used in any environmental education context e.g. environmental science classes at the high school level. In this game, a player plays the role as a scientist and investigates the health problems in their city. Students learn basic investigative skills (observation, hypothesis testing, data gathering, data analysis, and data reporting) that are a part of any environmental education curriculum.

”**UniGame: Social Skills and Knowledge Training**” is a framework that provides a possibility for every interested teacher to apply game-based learning for his/her classes. It can be classified as a role-play game, that fosters participation in

problem solving, effective communication, teamwork, project management, as well as other soft skills such as responsibility, creativity, micro-entrepreneurship, corporate culture, etc. The game is based on constructivist learning approach and collaborative learning. Using this game platform, teachers can define various topics, thus modifying the game for their own purposes.

TopSIM (2002) by TERTIA Edusoft provides different business games which have been used in business education and advanced training. A business game provides a model of the enterprise or of parts of the enterprise. The participants learn through experience the connections within the organization and the internal and external factors that influence the profit of the enterprise. They learn to implement business methods and means of information and to deal with uncertainty during decision making. The target groups are as follows: senior and junior managers in business, administration and authority, employees in technical and scientific areas, who need business knowledge for their activities, students of business administration and industrial engineering and trainees.

More examples on game-based learning can be found in [9]. Information on various games suitable for educational purposes can be found in a Survey on online game-based learning [5].

Chapter 4

ITS Support for GBL

We have selected Crossword and Treasure-Hunt as the games for our ITS framework. These games are selected as these can be placed in multiple subject areas. More games can be later added to our ITS framework. The instructor adds courses and topics to be taught by the ITS. For each topic or course, he can select a game as a pedagogical approach. The student has to select the course and topic he wants to learn. If the pedagogical approach for the topic is Game-based learning then the game will be presented in front of him. The main goal of the student should be to finish the game, learning happens involuntarily through incidental learning. Finally, feedback is provided to the students based on his performance. More details about each game and the steps to be followed by instructor, student and ITS are given below.

4.1 Crossword

The Crossword generation problem is: given an ordered finite alphabet A , a 0,1-matrix (grid), and a dictionary (sorted list of words of length no less than 2 over the alphabet A), replace all 1 entries in the grid with elements of A such that a crossword with (valid) dictionary words is obtained [2]. In general, there might be none, one or more solutions. Crossword generation involves different areas of interest in Artificial Intelligence, such as tree search, heuristics, planning, machine learning and natural language processing.

Crossword puzzles are used everyday by millions of people for entertainment, but have applications also in educational contexts. Infact, crossword puzzles are a fun and interactive way to test your knowledge of a particular subject. It is an excellent way to engage students in thinking laterally and building links in language. For example, some studies claim that simple thematic puzzles are amongst the most powerful tools to support the process of student learning and crossowrds can be a powerful tool to reinforce terminology and improve the learning of definitional items.

Practical applications of using and solving crosswords are many, including educational purposes such as learning words, spelling, languages and glossaries for specific

subjects. Instructors can create crosswords to assist their students in self assessing their knowledge of particular topics and in building up the required knowledge in order to successfully complete the course.

4.1.1 Existing Softwares

For the introduction of crossword puzzles into the pedagogical environment, it is desirable to have available generation engines which are freeware, or free of charge for usage in academic institutions. The bulk of the available off-the-shelf crossword puzzle generators are not free, and most of them are shareware, where the user is obligated to pay a registration fee after a specific trial period, typically 30 days. For instance, at the TUCOWS website listing Crossword Puzzle Generators, of the eleven titles presented, only one is freeware, while the others are either shareware or demo versions of commercial software titles.

Thus, the number of freeware titles devoted to crossword puzzle generation is limited. And the problem with the freeware generation engines is that they are standalone products and cannot be integrated into any other systems. Thus, they are of no use to our ITS framework.

Also, the academic research on crossword is done on the default crossword generation process where the grid is fixed. They focussed mainly on developing fast and effective search algorithms to find out the words which fit the grid.

4.1.2 Our Model

Our model is different from the general crossword generation problem where the grid is fixed and the number of words is very large. We need to build the crossword from a fixed set of words or answers given by the instructor. The grid varies depending on the input words. Infact, multiple crossword puzzles can be generated using a set of input words. There is also an option to select the grid size and the difficulty of the puzzle.

How is the difficulty of the puzzle determined? The difficulty level of the puzzle depends on two factors. Set of clues and the ability to deduce the answer from the grid. For a generic crossword puzzle, difficulty level of the puzzle is decided by the set of clues used i.e on how difficult it is to find the answers based on clues. But in our model we do not have control on the clues given so the difficulty mainly depends on the grid pattern. For example, guessing the answer is easier when we know the first character hence, having a link at the first character decreases the difficulty of the puzzle.

4.1.3 Instructor's Role

The instructor has to add questions and answers to the crossword module and then click on the generate crossword option. A crossword puzzle is generated based on the given words. Instructor has the option to select the grid size and also the difficulty level of the crossword puzzle. The difficulty depends on the number of interlinks and also on the position of the links. The grid size can be either 15x15 or 20x20.

4.1.4 Student's Role

Student fills in the words directly within the puzzle. The interface is designed such that whenever he fills a box, the cursor automatically goes to the next box. For a better user experience, the boxes where the answer should be entered is highlighted when the user clicks on the question. Once the student has completed his attempt at the puzzle, he can submit the puzzle.

4.1.5 Database Tables

Column Name	Type	Details
quiz_id	int(20)	Primary key
quiz_name	varchar(100)	name of the puzzle
grid	int(10)	15(15x15) or 20(20x20)
difficulty	int(10)	1-easy; 2-medium; 3-difficult
num_ques	int(10)	number of questions

Table 4.1: Crossword table

Column Name	Type	Details
quiz_id	int(20)	Foreign key
word_id	int(10)	Id of question - (quiz_id,word_id) makes the Primary Key
question	varchar(255)	question or clue
answer	varchar(50)	answer
x_val	int(10)	x co-ordinate of first character in the grid
y_val	int(10)	y co-ordinate of first character in the grid
axis	int(10)	used for orientation - 0 for horizontal and 1 for vertical

Table 4.2: Questions Table

Column Name	Type	Details
student_id	varchar(50)	Foreign key of student
quiz_id	int(20)	Foreign key of puzzle
word_id	int(10)	Foreign key of question
answer	varchar(50)	answer given by student

Table 4.3: Student Table

4.2 Scavenger Hunt

Treasure or Scavenger Hunt is an inquiry activity in which one systematically seeks knowledge with questions. Treasure is considered to be information, truth, or knowledge, and hunt implies inquiry, which is a systematic investigation. Treasure hunt activities are an engaging and active way for learners to enhance their skills.

Bates (1995) stated that the Scavenger Hunt game can produce significant savings in academic time, once designed, and greatly improve learning effectiveness.

Learners will be developing many generic skills as they complete the game. In particular they will be developing skills in: collecting, analyzing and organizing information, problem solving, self managed learning and using technology.

Why scavenger hunts are popular in academic area? Scavenger hunts, or treasure hunts, have quickly become one of the most popular tools for teaching students how to access and use the resources and information available on the Internet. There are many reasons for the hunts' rapidly growing popularity. Among them:

- Online scavenger hunts are easy to create and the resulting interactive searches are both fun and informative for students.
- The hunts can be geared to virtually any curriculum area, simultaneously providing students with technological and subject matter knowledge.
- Online scavenger hunts can be used as a whole class activity, as a team activity, or as a means of providing individual students with review or challenge activities.
- Scavenger hunts can be as simple or involved as circumstances dictate. Younger students may be provided with only a few questions, along with the links or URLs necessary for finding the answers, while older students may be given only a broad topic and asked to find their own sources for obtaining necessary information.

4.2.1 Existing Hunts

Below is a sampling of Internet scavenger hunts that have appeared on the pages of Education World during the past year [14]:

- **The Ants Go Marching**

Education World offers a scavenger hunt to capitalize on students' fascination with insects. This hunt has questions for students in all grades with an answer key also.

- **U.S. History of the 1930s**

This hunt encourages students to search the Web to learn more about the stories behind the people, places, events, and trends of the 1930s. Some of these include the Empire State Building, Superman, the Great Depression, Franklin Delano Roosevelt, the Monopoly board game etc.

- **Animals of the World**

To celebrate National Zoo and Aquarium Month, an online scavenger hunt is designed for the students. By exploring this hunt students will learn about the natural habitats of the animals. Two versions are designed, one for beginners and one for experienced surfers.

- **Track the Meteor Storm**

This scavenger hunt is created to help students understand the causes and effects of meteors and meteor showers. Three different scavenger hunts are created for elementary, middle, and high school students. Each scavenger hunt also includes several cross-curricular activities that can be used to extend the lesson.

A large collection of Scavenger Hunts is also available here [15].

4.2.2 Instructor's Role

The instructor has to first enter the number of levels for the scavenger hunt. Having different levels helps in managing the difficulty of the puzzle. The student can access a level only when he answers all the questions in the previous levels. The next step is to add questions and answers to the hunt. For each question he has to mention what level the question belongs.

4.2.3 Student's Role

The student starts the hunt with level 1. He can see all the questions in level 1 and can only reach the next level if he answers all the questions in this level. Students have to access and use the information available on the Internet or any other resource to find the answers. This develops the self managed learning and problem solving skills in the learner.

4.2.4 Database Tables

Column Name	Type	Details
th_id	int(20)	Primary key
th_name	varchar(100)	name of the hunt
no_levels	int(10)	number of levels

Table 4.4: Scavenge Hunt table

Column Name	Type	Details
q_id	int(20)	Primary key
question	text	question
answer	varchar(255)	answer
th_id	int(20)	Foreign key
level	int(20)	level the question belongs

Table 4.5: Questions Table

Column Name	Type	Details
user	varchar(100)	Foreign key of student
th_id	int(20)	Foreign key of hunt
level	int(20)	level of hunt
start_time	varchar(100)	Time when student entered the level
end_time	varchar(100)	Time when student answered all questions in the level

Table 4.6: Student Level Table

Column Name	Type	Details
user	varchar(100)	Foreign key of student
q_id	int(20)	Foreign key of question

Table 4.7: Attempted Questions Table

4.3 Steps to be followed by Instructor

1. The instructor has to login first using his username and password.
2. Select or create a course
3. Select or create a topic for the course
4. Select the game which he wants to use as a teaching technique
5. For Crossword:
 - a. Add questions and answers to the game
 - b. Select the difficulty level of the puzzle
 - c. Click on the generate puzzle

6. For Treasure-Hunt:
 - a. Select number of levels
 - b. Add questions and answers to the game
 - c. Arrange the questions into different levels
7. After the students played the game check the students analysis report and give feedback

4.4 Steps to be followed by Student

1. The student also has to login first using his username and password.
2. Select Course
3. Select Topic and sub-topic
4. Play the game which are provided under the given topic and course
5. Check the answers and learn from mistakes
6. Check for feedback later from instructor

4.5 Steps to be followed by ITS

1. Allow login of users (teachers and students)
2. Maintain a profile for both teachers and students
3. For teachers, provide a interface for
 - a. creating/selecting course and topic
 - b. designing the game
 - c. providing results and feedback at the end of game
4. For students, provide a interface for
 - a. selecting course and topic
 - b. playing the game
 - c. viewing results and feedback after the game play
5. Maintain a student performance table
6. Get feedback from teacher after the completion of game

4.6 Crossword Examples

The instructor's interface when he tries to create a puzzle.

The screenshot shows the 'INTELLIGENT TUTORING SYSTEM' interface. At the top, there is a dark blue header with the system name in white. Below the header is a navigation bar with links for HOME, STUDENT, INSTRUCTOR, ABOUT, LINKS, and CONTACT. The main content area is divided into two columns. The left column is titled 'ADD QUIZ' and contains a form with the following fields: 'Quiz Name:' with a text input field; 'Grid:' with radio buttons for '15x15' (selected) and '20x20'; 'Difficulty:' with radio buttons for 'Easy' (selected), 'Medium', and 'Hard'; 'Num of Questions:' with a dropdown menu; and a 'Submit' button. The right column is titled 'WELCOME XYZ' and 'YOU CAN DO', and contains a list of actions: 'Add Exercise', 'Edit Level Limit', 'See Student's Progress', and 'Upload Resources'.

Figure 4.1: Add Quiz

When instructor adds the number questions, the question field appears automatically without refreshing the page.

INTELLIGENT TUTORING SYSTEM

HOME STUDENT INSTRUCTOR ABOUT LINKS CONTACT

ADD QUIZ

Quiz Name:

Grid: 15x15 20x20

Difficulty: Easy Medium Hard

Num of Questions : ▼

Question 1:
Answer:

Question 2:
Answer:

Question 3:
Answer:

Question 4:
Answer:

Question 5:
Answer:

WELCOME XYZ

YOU CAN DO

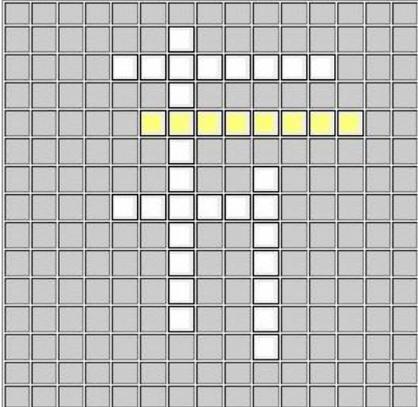
- [Add Exercise](#)
- [Edit Level Limit](#)
- [See Student's Progress](#)
- [Upload Resources](#)

Figure 4.2: Add Quiz 2

Students interface. When he clicks on a question, the boxes where he has to answer are highlighted.

Difficulty: Easy

Across	Down
Which is the statement when called terminates the execution of a function and returns control to the calling function?	The mechanism of reusing and extending existing classes without modifying them, thus producing hierarchical relationships between them.
Feature of the C++ programming language that allow functions and classes to operate with generic types	A type of variable to store a determined value
These are used to structure our programs in a more modular way	



WELCOME XYZ

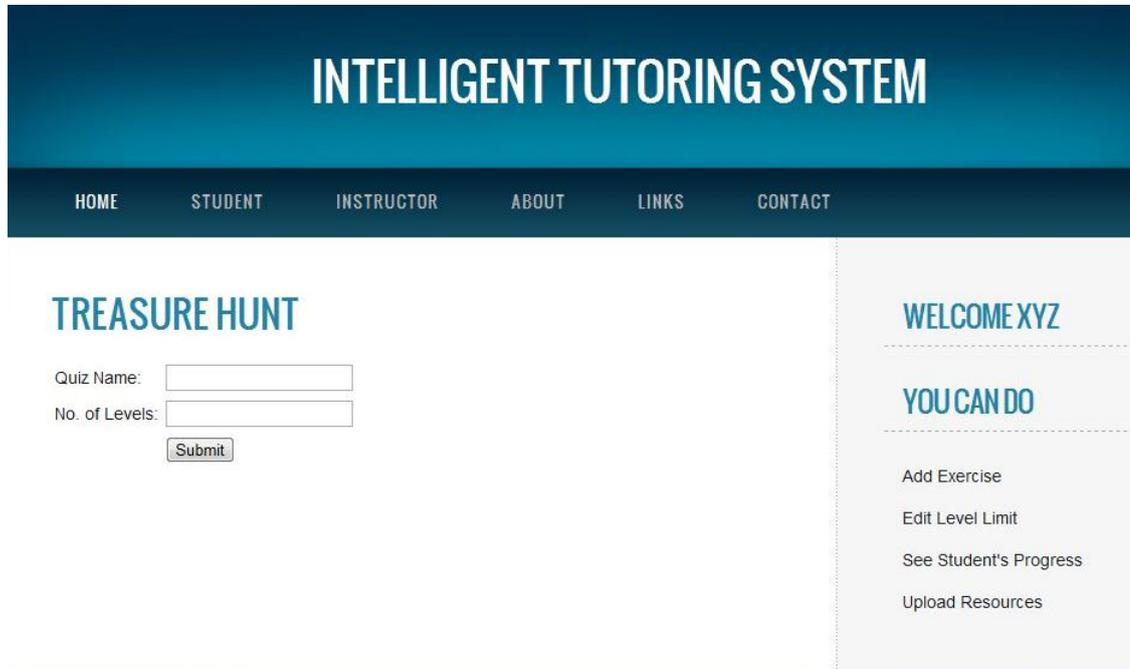
YOU CAN DO

- [Attempt Exercise](#)
- [See your Level](#)
- [See Your Progress](#)
- [Available Resources](#)

Figure 4.4: Student interface

4.7 Scavenger Hunt Example

The instructor's interface when he tries to create a Scavenger hunt puzzle.



The screenshot shows the instructor interface of an Intelligent Tutoring System. At the top, a dark blue header contains the text "INTELLIGENT TUTORING SYSTEM" in white. Below this is a navigation bar with links for "HOME", "STUDENT", "INSTRUCTOR", "ABOUT", "LINKS", and "CONTACT". The main content area is divided into two sections. On the left, under the heading "TREASURE HUNT", there are two input fields: "Quiz Name:" and "No. of Levels:", each followed by a text box. Below these fields is a "Submit" button. On the right, there is a sidebar with a "WELCOME XYZ" message, followed by a "YOU CAN DO" section containing a list of actions: "Add Exercise", "Edit Level Limit", "See Student's Progress", and "Upload Resources".

Figure 4.5: Start Hunt

The interface where questions are added

The screenshot shows a web interface for an Intelligent Tutoring System. At the top, a dark blue header contains the text "INTELLIGENT TUTORING SYSTEM" in white. Below this is a navigation bar with links for "HOME", "STUDENT", "INSTRUCTOR", "ABOUT", "LINKS", and "CONTACT". The main content area is titled "TREASURE HUNT" and contains a form for adding questions. The form includes a "Question:" label followed by a large text input field, an "Answer:" label followed by a smaller text input field, a "Level:" label followed by a dropdown menu, and a "Submit" button. On the right side of the page, there is a sidebar with a "WELCOME XYZ" message and a "YOU CAN DO" section containing links for "Add Exercise", "Edit Level Limit", "See Student's Progress", and "Upload Resources".

Figure 4.6: Add Questions

The instructor can also view the list of questions

INTELLIGENT TUTORING SYSTEM

HOME STUDENT INSTRUCTOR ABOUT LINKS CONTACT

TREASURE HUNT

LEVEL 1

1. Which is the statement when called terminates the execution of a function and returns control to the calling function?
Ans: return

2. These are used to structure our programs in a more modular way
Ans: Function

LEVEL 2

1. Feature of the C++ programming language that allow functions and classes to operate with generic types
Ans: Template

2. The mechanism of reusing and extending existing classes without modifying them, thus producing hierarchical relationships between them.
Ans: Inheritance

LEVEL 3

1. Predict the output or error(s) for the following:

```
main()
{
char s[ ]="man";
printf("%c%c%c%c",s[0],*(s+1),*(1+s),2[s]);
}
Ans: maan
```

WELCOME XYZ

YOU CAN DO

- Add Exercise
- Edit Level Limit
- See Student's Progress
- Upload Resources

Figure 4.7: List Questions

Students interface.

The screenshot displays the user interface of an Intelligent Tutoring System. At the top, a dark blue header contains the title "INTELLIGENT TUTORING SYSTEM" in white, bold, uppercase letters. Below the header is a navigation bar with six menu items: HOME, STUDENT, INSTRUCTOR, ABOUT, LINKS, and CONTACT, all in white, uppercase letters. The main content area is divided into two columns. The left column features two levels of exercises. "LEVEL 1" includes two questions: "1. Which is the statement when called terminates the execution of a function and returns control to the calling function?" with the answer "return", and "2. These are used to structure our programs in a more modular way" with the answer "Function". "LEVEL 2" includes two questions: "1. Feature of the C++ programming language that allow functions and classes to operate with generic types" with an empty answer box and a "Submit" button, and "2. The mechanism of reusing and extending existing classes without modifying them, thus producing hierarchical relationships between them." with an empty answer box and a "Submit" button. The right column is a sidebar with a light gray background. It starts with "WELCOME XYZ" in blue, uppercase letters, followed by a dashed horizontal line. Below the line is "YOU CAN DO" in blue, uppercase letters, followed by another dashed horizontal line. Underneath are four links: "Attempt Exercise", "See your Level", "See Your Progress", and "Available Resources", all in a standard blue font.

Figure 4.8: Student interface

Chapter 5

ITS Framework and Workflow

5.1 Components of Intelligent Tutoring Systems

The architecture used in our ITS is very similar to the architecture presented by Burn and Caps[1]. The architecture was based on four main components. These four modules are represented as the domain module, student module, pedagogical module and the interface module.

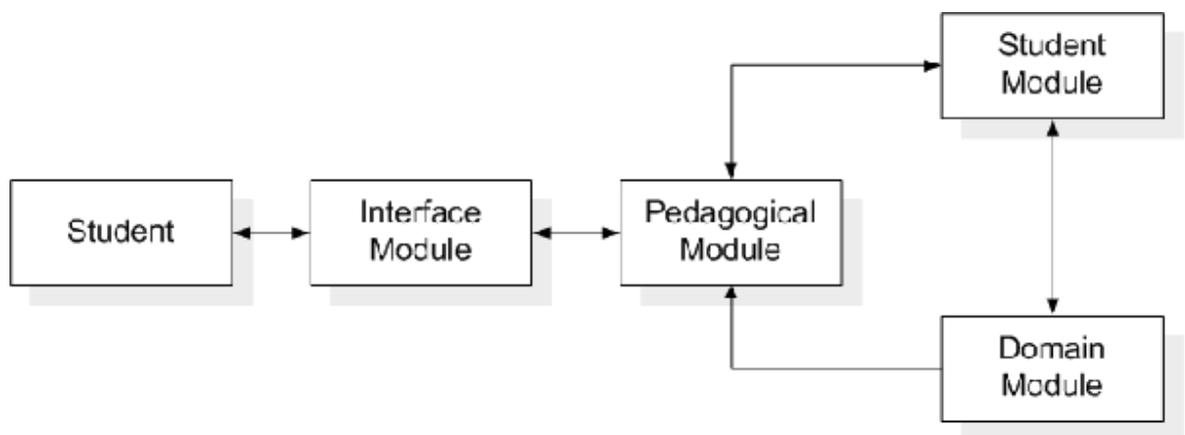


Figure 5.1: Components of ITS

5.1.1 Domain Module

The domain module is the knowledge management system and represents the content knowledge that the student is acquiring. All the concepts that the system pretends to transmit to the student are stored in this module. This module is at the heart of an intelligent tutoring system and provides the basis for interpreting student actions.

5.1.2 Student Module

The student module is a record of the student's knowledge state. It stores information that is specific to each individual learner. All the student behavior is recorded in the system and used for "reasoning" and adapt the domain module to the learner's needs. According to Wenger, student models have three tasks.

- They must gather data from and about the learner. This data can be explicit - asking the student to solve specific problems - or implicit - tracking the students navigation and other interactions and comparing them to information about similar learner responses.
- They must use that data to create a representation of the student's knowledge and learning process. The system then uses this model to predict what type of response the student will make in subsequent situations, compares that prediction to the students' actual response, and uses that information to refine the model of the student.
- They should select optimal pedagogical strategies for presenting domain information to the student based on student's knowledge and learning process.

5.1.3 Pedagogical Module

The pedagogical module provides a model of the teaching process. It acts as a virtual instructor, presenting the contents in an appropriate sequence, based on the students knowledge and his learning style. This is an interactive process and this module has the mission to explain the concepts to the student given several points of view and supporting all the learning process. For example, information about when to review, when to present a new topic, and which topic to present is controlled by the pedagogical module. As mentioned earlier, the student module is used as input to this component, so the pedagogical decisions reflect the differing needs of each student.

5.1.4 Interface Module

The interface module communicates and interacts with the student. It controls the dialogue and the screen layouts of the system. To develop a good interface module it is necessary to consider the usability issues of a user computer interface, because this module interacts with the user and the other components of the system. If the interface fails all the other modules fail too.

5.2 Architecture

The overall architecture diagram of our ITS Framework is given below. The most important part of the architecture is the controller which belongs to the Pedagogical Module. Whenever a student interacts with it, it access the Student Module for student profile and Domain Module for content information and selects the appropriate teaching strategy or quiz based on his learning capability.

The overall content management is performed by the instructor using the Domain Module. He can manage the course structure and also add content to the question bank.

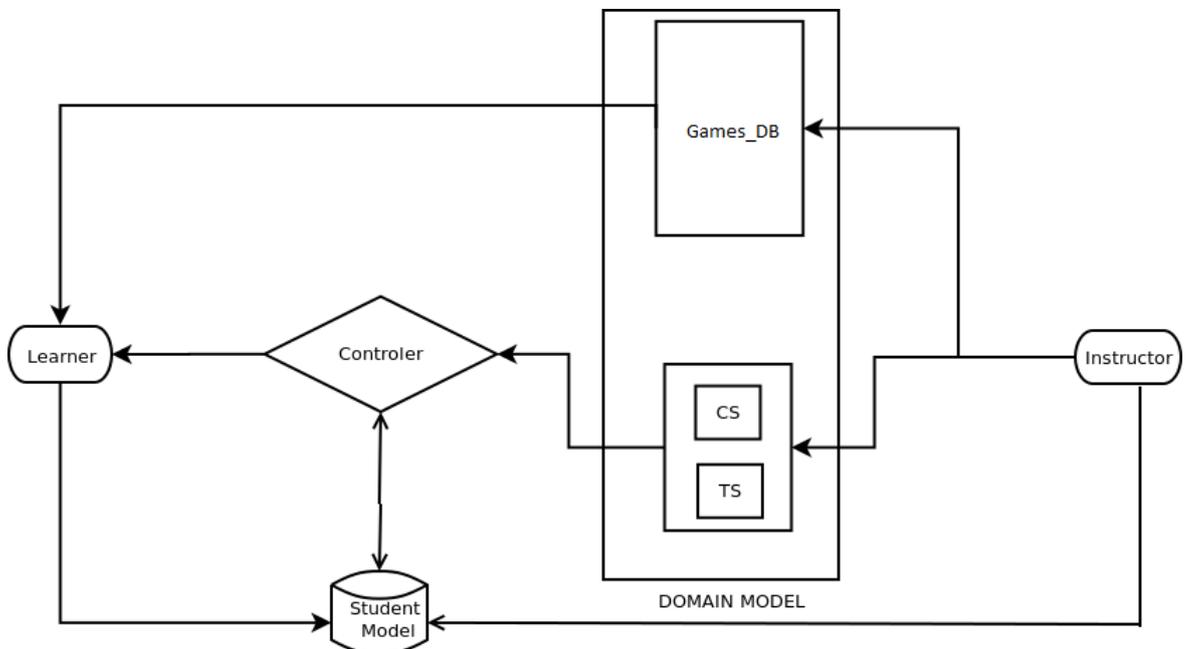


Figure 5.2: Architecture of our ITS

5.3 Time Sequence Diagrams

5.3.1 Time Sequence Diagram For Instructor

The below figure shows what are the steps needed to be done by the instructor. The controller is a part of the pedagogical module which decides what teaching strategy to use for a particular topic. The instructor uses this controller to add questions to the teaching strategies. The controller provides different interface to each teaching strategy since their requirements are different. To reach to this part, the instructor has to first login and then select course, topic and sub-topic for which he wants to create the quiz.

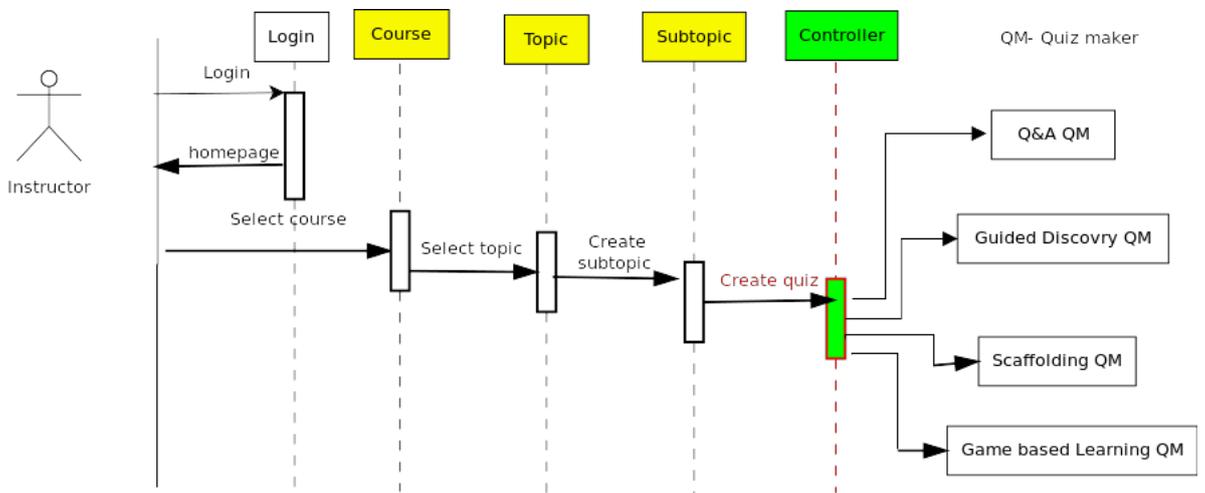


Figure 5.3: Time sequence Diagram for Instructor in Our ITS

5.3.2 Time Sequence Diagram For Student

The below figure shows how the student interacts with the system. For attempting an exercise, he has login first like the instructor and then select course, topic and sub-topic he wants to learn. The controller then brings up the appropriate quiz into the front. If there are multiple teaching strategies available for a single sub-topic then the teaching strategy is decided based on the learning ability of the student.

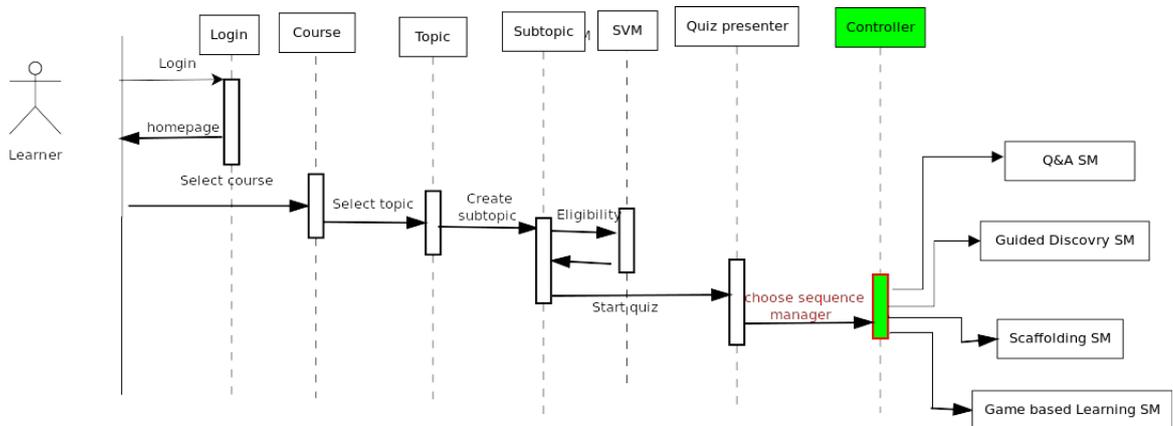


Figure 5.4: Time sequence Diagram for learner in Our ITS

5.4 Modules

This section describes the different modules that are part of our ITS Framework. The functionality and use of each module is explained clearly. The different modules used by our ITS are given below.

- GUI Module
- Input Validation module
- Registration Module
- Authentication module
- Course Module
- Topic Module
- Game Module
- Database Handling Module
- Log module
- Feedback module

5.4.1 GUI Module

The GUI module handles the display interface of the users. The important functionality of the GUI module is that the user experience should be flawless. Important care is taken in making the interface as rich and intuitive as possible.

5.4.2 Input Validation module

This module validates all the user input data. The various functionalities of this module are:

- Check whether all the mandatory input data is filled
- Check whether input is in proper format
- Check for unwanted scripts in user submitted data

5.4.3 Registration Module

This module is used to add users(students and instructors) to our database. The users can register themselves to our interface or the admin adds them manually.

5.4.4 Authentication module

This module authenticates the validity of a user. The user has to login using his username and password. Only when he enters the correct information he is allowed to access the system.

5.4.5 Course Module

This module maintains the course structure in the Domain Model. It has following two functionalities:

- Allows instructor to maintain the course structure. He has permissions to add and manages courses
- Allows student to access the course content

5.4.6 Topic Module

The functionality of this module is similar to the Course Module.

- Allows instructor to manage the topic structure for a particular course
- Instructor can create a dependency list for a topic and a student can access the topic only if he learns all the topics in the dependency list.
- Allows student to access the topics and sub-topics of a course

5.4.7 Game Module

This is the core module which maintains the game structure. The instructor accesses this to create games and students to play the games provided by the instructor.

5.4.8 Database Handling Module

This module handles all the database related operations like insert, delete, update, select etc. The following are the main databases which we use:

- User Database - Contains the login and personal information of every user
- Domain Database - Contains course and topic information
- Game Database - Contains all the information regarding games like questions, answers and order etc.
- Log Database - Contains all the user activity of every student

5.4.9 Log module

This module logs all the user actions. This is mainly used to track students actions so that the instructor can give feedback based on this.

5.4.10 Feedback module

This module is used to present the feedback to the student after playing a game so that he can access his performance. The instructor can also give his feedback seeing the student performance table later.

Chapter 6

Android App

6.1 Why there is a need for mobile app?

India has undergone a mobile phone revolution over the past few years. With over 771 million mobile lines in service, India now boasts of being the second largest telecommunication network in the world after China. As the proliferation of smartphones continues to rise among 15- to 24-year-olds, coupled with the fact that there were over 22m app downloads in India last year, the education sector has to finally embraced mobile into their digital strategy to enhance communication with current and prospective students. There has been an increasing use of smart phones by college students. For many Indian students in the cities and rural areas alike, smartphones will serve as PCs. If smartphones are not in the hands of a college student just yet, they soon will be. These students spend hours a day on sites like Facebook and YouTube, play highly immersive video games and interact with apps on their mobile devices. We are wasting the huge opportunity offered by technology to engage and immerse students in learning activity. A mobile app brings the learning process into the hands of student as they access it at anytime.

6.2 Current Apps for Education

- **Blackboard Mobile Learn**

Blackboard is a very popular online learning and assignment-sharing tool. Many universities and high schools in US are using Blackboard to help students and teachers stay connected outside of the classroom. Having access to Blackboards Mobile Learning app will give you access to the service at all times, so that you are always in the know.

- **eClicker**

eClicker is a wonderful app that allows the students to interact with their lessons as they go. Students will use their devices to answer multiple choice questions and respond to polls that are incorporated into the lesson. This adds a nice level of interactivity that will keep your students engaged and on their toes.

- **CourseSmart**

CourseSmart is a mobile textbook app that brings almost every textbook out there instantly to the pocket of the student. CourseSmart works with most major publishers to ensure that your textbook can follow you wherever you go, without all the nagging back pain that comes with the hardcover version.

6.3 Activities in Android

The user interface of any android application is displayed on a device through an Activity. Internally there is a stack of Activities and typically, one Activity is created for each unique screen. When moving from one screen to another, the next Activity to be visible is pushed onto the top of the stack i.e. the Activity on the top of the stack is what is visible on the display. Activities are popped from the stack by pressing the back button, which resumes the previous Activity.

The following are the Activities built for this app:

- LoginActivity
- RegisterActivity
- CourseListActivity
- TopicListActivity
- SubtopicListActivity
- ScaffoldingActivity
- SocraticActivity
- GuidedDiscoveryActivity
- CompileActivity
- CrosswordActivity
- TreasurehuntActivity

6.4 Layouts

Layout is a type of View class whose primary purpose is to organize other controls. It defines what is drawn on the screen. Layout resources are stored as XML files in the `/res/layout` resource directory for the application. A layout resource is simply a template for a user interface screen, or portion of a screen. These layout classes (`LinearLayout`, `RelativeLayout`, `TableLayout`, etc.) are used to display child controls, such as text controls or buttons or images on the screen. Android user interfaces can

be defined as layout resources in XML or created programmatically.

The following are the layouts used:

- login.xml
- register.xml
- list.xml
- list2.xml
- quiz.xml
- compile.xml
- crossword.xml
- treasurehunt.xml

6.5 PHP API classes

An android app cannot directly access the database on an external server. PHP APIs which are hosted on the external server acts as an intermediate between the app and database. It accepts requests from app, gets the required data from the database and outputs it back to the app. PHP API classes built does the following functionalities:

- Accept requests by GET/POST methods
- Interact with PHP classes to get data from database or store in database
- Return output in JSON format

Following are the include files required to build API in php.

- **config.php** - This file contains constant configuration variables like database connection strings and other constants. These variables are used to establish a connection with the database.
- **DB_Connect.php** - A class contains methods to connect or disconnect from the database.
- **DB_Functions.php** - A class contains methods to insert/read operations on database. This file contains functions to store user in database, get user from database etc. You can also add methods like update user, delete user.

Below are the files that handle all the requests. These files accepts all GET and POST requests. They play role of accepting requests and giving response. On each request they will talk to database and will give appropriate response in JSON format.

- index.php - It handles the requests for login and registration of users
- course_list.php - This file sends the course list as a response once the user login
- topic_list.php - This file handles the request for topic list of a course as well as the sub-topic list
- scaffolding.php - Handles the requests of Scaffolding teaching strategy
- socratic.php - Handles the requests of Socratic teaching strategy
- guided.php - Handles the requests of Guided Discovery teaching strategy
- treasurehunt.php - Handles the requests of treasure hunt game
- crossword.php - Handles the requests of crossword

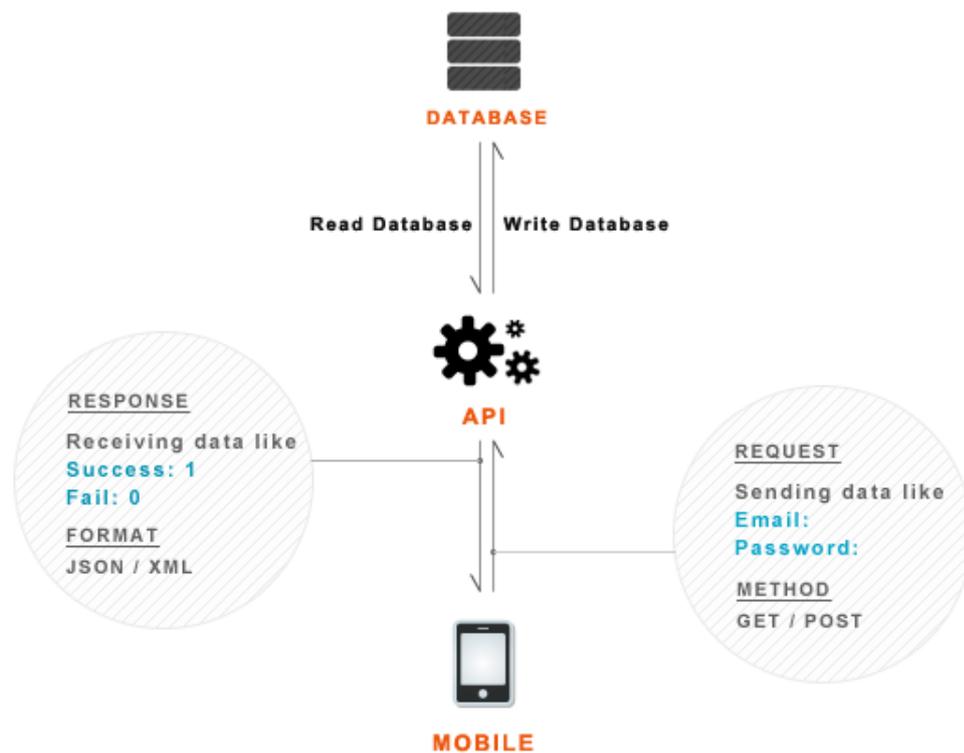


Figure 6.1: PHP API Working [17]

6.6 JSON Responses

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for both humans to read and write and also for machines to parse and generate. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. Earlier web services used XML as their primary data format for transmitting back data, but since JSON appeared, it has been the preferred format because it is much more lightweight.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

6.6.1 Types of API JSON Responses

The following is an example of the type of JSON responses generated by API.

Login Success Response - Success Code = 1 (User Logged in)

```
{
  "tag": "login",
  "success": 1,
  "error": 0,
  "uid": "2",
  "user": {
    "name": "Praveen",
    "email": "praveen@gmail.com",
  }
}
```

Login Error Response - Error Code = 1 (Login Error Incorrect username/password)

```
{
  "tag": "login",
  "success": 0,
  "error": 1,
  "error_msg": "Incorrect email or password!"
}
```

Chapter 7

Conclusion and Future Work

In the first chapter of the report, it is mentioned that the ultimate goal of the project is to have a system with minimal human intervention in which students can learn and get instant feedback and hints. This project is only a first step towards achieving that goal. The modular architecture ensures that each module can be extended and developed independently. This opens up a lot of avenues for future research and development.

7.1 User Interface

The interfaces are built only for the prototype and needs to be developed more. More effort is required towards creating a flawless user experience for both students interface and instructors interface.

7.2 Evaluation

The system must be evaluated from both a design perspective and from an Educational Technology perspective. Evaluation must be done for each module and also for the whole integrated system. The interfaces must be evaluated for the user experience. As mentioned earlier, a flawless user experience is crucial for the success of the system as an effective learning tool because a bad interface could lead to cognitive overloading of the student's working memory.

Over all, the effectiveness of the system as a learning tool should also be evaluated either in a laboratory setting with a control group or by subject experts.

7.3 Expanding the Scope

At this time we have a basic framework for developing ITS. There is lot more remains to be done. One can develop an ITS according to ones requirement. In future following thing can be done.

- **Collecting material for teaching:** The very first work is to prepare the material to student for learning like question with hints and some short tricks etc.
- **Response Time Theory:** One can implement a module for time response. So that time taken by student in attempting a question or subtopic can be stored. This stored time then can be used to take important decisions. This time will be helpful to find out the interest of student in topic or learning.
- **More teaching strategies:** Since we followed a modular architecture, more teaching strategies can be easily added.
- **More Games:** The present games are selected based on no graphics requirement. The graphics can be improved later and games which require higher visuals can also be added.
- **Collaborative learning:** Some module for providing collaborative learning also can be added to the system. So that students can interact with each other and learn more with each others experience. The level of student can be used as a competitive factor by showing top level students to each student.

Bibliography

- [1] Burn and Caps, *Intelligent Tutoring Systems*, Lawrence Erlbaum Associates, Hillsdale, NJ., 1988.
- [2] Ariel Arbiser, *Practical Crossword Generation with Checkpoint Search*, (Matrahaza, 1995), University of Buenos Aires Ciudad Universitaria, Argentina, 2001, pp. 1–6.
- [3] Corbett A.T., Koedinger K.R., & Anderson J.R., *Intelligent tutoring systems*, In M.G. Helander, T.K. Landauer, & P.V. Prabhu (Eds.), Handbook of human-computer interaction, 1997, pp. 849874.
- [4] J. Anderson, C. Boyle, R. Farrell, & B. Reiser, (1987). *Cognitive principles in the design of computer tutors*. In P. Morris (Ed.), Modeling cognition. NY: John Wiley.
- [5] Dondi C., Moretti M.: *Survey on online game-based learning*. Retrieved 18. 08. 2003, from http://www.unigame.net/html/case_studies/D1.pdf
- [6] Vikash Kumar, *Development of an Intelligent Tutoring System Framework for Socratic Questioning*, M-Tech Thesis, IIT Bombay 2012
- [7] Chandrapal Singh, *Development of an Intelligent Tutoring System Framework for Scaffolding*, M-Tech Thesis, IIT Bombay 2012
- [8] M. Rajashekhar, *Development of an Intelligent Tutoring System Framework for Guided Discovery*, M-Tech Thesis, IIT Bombay 2012
- [9] Prensky, M. (2001). *Do They Really Think Differently?*. In On the Horizon, December 2001, 9 (6). Lincoln: NCB University Press.
- [10] M. Spectre, and M. Prensky (2001), *Theoretical Underpinnings of Games2Train.com's approach*, Games2train, 2001.
- [11] Garris, R., Ahlers, R., & Driskell, J. E., *Games, motivation, and learning, Simulation & Gaming*, An Interdisciplinary Journal of Theory, Practice and Research. Vol33, No.4 Dec. 2002, pp. 441-467.
- [12] Paul R Kearney, Maja Pivec, *Recursive Loops of Game-Based Learning: a Conceptual model*.

- [13] Dong Won Kim, JingTao Yao, *A Treasure Hunt Model for Inquiry-Based Learning in the Development of a Web-based Learning Support System*, Journal of Universal Computer Science, vol. 16, no. 14 (2010)
- [14] Scavenger Hunts: Searching for Treasure on the Internet, http://www.educationworld.com/a_curr/curr113.shtml
- [15] Scavenger Hunts, http://faculty.usiouxfalls.edu/arpeterson/scavenger_hunts.htm
- [16] Android Developers, <http://developer.android.com/index.html>
- [17] Android hive, <http://www.androidhive.info/>