International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-11 Issue-7, June 2022

Designing and Implementing Accommodation Management System: ASHAMS as Case Analysis

Ronok Bhowmik, Md. Hasnat Riaz

Abstract*: Most Bangladeshi schools, colleges and universities rehearse the old conventional and manual accommodation management procedures. Accommodation management in a manual way is a tedious paperwork process since it involves unnecessary time consumption and lots of unwanted errors. This manual procedure lingers the seat management process (allocation-deallocation, room shifting (reallocation), etc.) and slows down the overall work speed for both the hall managerial bodies and students. We have explored the feasibility studies and requirement analysis considering all the manual accommodation management processes. We have proposed and designed a web-based Abdus Salam Hall Accommodation Management System (ASHAMS) according to the outcome obtained. Tools used to implement the system are Microsoft Visual studio and ASP.net framework as the front-end and SQL Server as the back-end server database. We proposed ASHAMS as a pilot project, and further implementation depends on the success of this project. We collected data from Bhasha Shahid Abdus Salam Hall, Noakhali Science and Technology University (NSTU), Bangladesh, for the entire development purpose of ASHAMS. Using ASHAMS, Hall (dormitory/hostel) authority can easily manage the hall details, room details, seat management process and reduce human errors. Hopefully, ASHAMS will overcome the shortcomings of conventional accommodation management procedures; improve the service quality, productivity, personnel efficiency, reliability, and transparency in the organization.

Keywords: Agile Methodology, Accommodation Management System, SQL Server, Use Case diagram.

I. INTRODUCTION

The accommodation management system could be a web/desktop-based application that reliably manages accommodation facilities. This kind of system includes various official and non-official operational tasks like seat allocation-deallocation, reallocation, keeping student and room details, capturing leaving information, seat rent-related information, fixed-asset details, and generating relevant reports. Schools, colleges, and universities maintain different logbooks to process all the activities over the years. This monotonous time-consuming paperwork deals with extremely clunky activities that often create a lot of unwanted human errors. Considering these, it is necessary to have an accommodation management system that is not only dynamic but also digital. A dynamic hall management system can reduce human efforts and errors and maximize the efficiency of the entire process. In addition, a dynamic hall management system can reduce the strain or stress on the responsible authorities, and malpractice can be lessened. NSTU is a science and technology-based university in Bangladesh. Currently, there are five halls of accommodation for the students that accommodate more than 4000 students. For the time being, the hall authorities announce seat allotment notice biannually. The seats are disbursed according to some benchmarks determined by the provost bodies and hall associate administrative staff. The provost body consists of the Provost and Assistant Provosts of a hall. Generally, In Bangladeshi universities, the provost is the amenable person for a specific hall. The purpose of this paper is to design a highly portable, GUI (Graphical User Interface) based and user-friendly ASHAMS that helps to (1) maintain and navigate the system smoothly, (2) insert, update, delete, and fetch information (present and past data, hall details, student details, provost details, guest and visitor records, hall rules and regulations) easily, (3) eliminate redundancy of data and wrong entry, (4) increase efficiency by reducing workload and human errors, and (5) provide reliability and security. Our manuscript is organized as follows: Section II is earmarked for analyzing previous related works. Section III acquaints the materials and methodology of our work. We depict results and discussions in section IV. In section V, we shorten our conclusions on the study, lessons learned, and provide future work direction.

II. RELATED WORKS

We reviewed several similar research projects. Most of the research projects deal with the entire hostel management system comprised of Laundromat management, dining management, seat management, and hall library management system. In [1], for designing the Dormitory Management system (DMS), the authors investigated the feasibility studies and functional requirements. Then they implement a DMS using the Java, SSM framework, and MySQL technology. The development procedures of an automated hostel facility management system were presented by the researchers of [2]. Here, developers developed it using Visual Basic and Microsoft Access.

Manuscript received on 23 May 2022.
Revised Manuscript received on 30 May 2022.
Manuscript published on 30 June 2022.

* Correspondence Author

Ronok Bhowmik*, Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University (NSTU), Noakhali, Bangladesh. Email: ronokbhowmik@admin.nstu.edu.bd

Md. Hasnat Riaz, Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University (NSTU), Noakhali, Bangladesh. Email: hasnat.cste@nstu.edu.bd

It also contains an incorporated authentication algorithm for preventing unwarranted access. In article [3], the authors described the development of a dormitory management system established on agile development architecture. Reference [4] explores the issues and challenges encountered by several universities in Indonesia in enrolling in dormitories either online or offline. Article [5] illustrated the university dormitory management system established on agile development architecture. The consequences of agile methodology on software processes according to the quality within the organizational, methodical, and cultural framework are described by the authors of [6]. The authors of [7] explain some prominent agile methodologies like Extreme Programming (XP), Feature-Driven Development (FDD), and Scrum. In [8] authors reviewed the modern Software Development Life Cycle (SDLC) and explained the merits and demerits of traditional and agile methodology. In addition, they suggest some areas of improvement for current Agile Development. Software development models, namely the Waterfall Model, Iterative Model, V-shaped Model, Spiral Model, Extreme Programming, Iterative and Incremental Method, Rapid Prototyping Model, the Chaos Model, Adaptive Software Development (ASD), the Agile Software Process (ASP), Dynamic System Development Method (DSDM), FDD, Rational Unified Process (RUP), SCRUM, Wisdom, and the Big Bang Model are discussed thoroughly by the authors of [9]. They make a comparison among these models. In addition, they reviewed the features and defects of each model. The authors of [10] discussed the advantages and disadvantages of some notable software development models. In article [11], the authors developed an administrative accommodation system by setting up and maintaining a backstage supporter's database. Article [12] depicted the alternative use of the smartphone as a substitute for the smart card technique. Near Field Communication (NFC) was used for developing a secured system and identical use to the smart card system. Automating the accommodation process at the "Julio Antonio Mella" headquarters at the Universidad de Oriente, theoretical methods such as historical–logical, analytical–synthetic, systemic–structural, and methodological foundations were exposed wisely [13]. The authors of [14] emphasized the security issues and their importance in SDLC. They also analyzed risk classification for better risk assessment.

III. MATERIALS AND METHODOLOGY

To design and develop ASHAMS, we have followed the SDLC. SDLC is a combination of some sets of procedures that lead us to develop a software product. It helps us design, create and construct high-quality, profitable, and trustworthy software products. In addition, it also ensures on-time product delivery. SDLC consists of (1) Planning stage, (2) Requirement analysis stage, (3) Design and Prototyping stage, (4) Software development (coding) stage, (5) Software Testing stage, (6) Implementation and Integration stage, (7) Operations and Maintenance stage. Here, we have concentrated on all the stages of SDLC according to the development of ASHAMS.

A. Planning stage

Planning is the first stage of any development work. No project will be successful without proper planning. Proper planning helps us to determine the problem and scope of the system. In software development, planning means the feasibility study. For the ASHAMS project, we have reviewed the feasibility study preferably. The Feasibility study, often known as Feasibility analysis, helps us determine whether a proposed project will be successful or not. The objective of feasibility study is to find an optimal solution that deals with improving the existing system, knowing what should be embedded in the new system, defining the problems and objectives, and avoiding costly repairs at a later stage when the system is implemented. The feasibility study is accomplished once the concerned territory is indisputably comprehended and determined how quickly to solve the problem using minimum expense. If we want to design and develop a new system, starting the process with a feasibility study is wise. Business requirements are the input for the feasibility study. In a feasibility study, we usually consider three inter-related types of feasibility. These are Technical, Operational, and Financial feasibility.

- Technical feasibility: The main concern of technical feasibility is to specify the software that will satisfy the user requirements. ASHAMS runs on any platform (machine) since the C# is considered platform-independent. It runs with minimum system requirements and system resources. New modules can be integrated later on the application if required. So, ASHAMS is technically feasible.
- Operational feasibility: The operational feasibility helps us specify the software that is easy to operate for the end-users. ASHAMS is GUI-based user-friendly software with simple instructions that require no special skills to operate the system. Unwanted human errors, unnecessary rework, and workload can be reduced through this system. New end-users will find it comfortable to use. So, ASHAMS is operationally feasible.
- Financial feasibility: Cost-benefit analysis, long-term co-operative income strategies, and the cost of resources needed for development and after development phases are the spectrum of Financial feasibility. In the manual process, hall authorities have to maintain several logbooks/registers. This monotonous task can be avoided by processing the data in a computerized format that is cheaper and more reliable. Since the cost of resources for the development and maintenance of the system satisfies the organization, ASHAMS is financially feasible.

B. Requirement analysis stage

Requirement Analysis is one of the most crucial parts of software development. Before execution, requirement analysis helps the developer to find the necessities of the system development. The requirements of a system are the descriptions of what the system can do, what kind of services are provided, and the constraints of that operation. The requirements of a system represent the need of a customer (system user). The requirements are often classified as User Requirements and System Requirements in the requirement engineering process.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

User requirements are high-level abstract requirements. In detail, we can say that user requirements are statements with natural languages and diagrams.

In contrast, System requirements are the precise report (functions, services, and operation constraints) of what the system can do or not. Software system specifications are often classified as Functional requirements and Non-functional requirements.

- SRS (Software Requirement Specification) Document: We collected actual data by arranging several interview sessions with the students, provost bodies, staff, and other associated persons of Bhasha Shahid Abdus Salam Hall. After data collection, we segregated the problems by user categories. We analyzed the functionalities and the outcomes of the feasibility study. After that, we prepared a document generally known as SRS. In the System specification, we prepared the system and software design. The output gathered from the requirement analysis works as the input for the system specification.

Lack of proper SRS document, the development process may turn into a chaotic implementation process, generate poor performance, and even turn into a total failure of the project. For skipping all of these scenarios, we must maintain the SRS document. SRS document describes every detail of the project, and it is the most critical document in SDLC. The system requirements (functional and non-functional) are specified in this phase. In addition, it illustrates the overall system architecture. In a word, SRS is the roadmap for the project. A proper SRS document helps the software developer to design and code properly. It minimizes the ratio of code refactoring after development. SRS also helps the investors to get a clear idea of cost estimation. The software tester gets the guidelines to design the test case properly. For ASHAMS, we highlight Functional and non-functional requirements as follows:

Functional requirements specify how software system behaves in particular conditions. We categorized the Functional requirements based on the two categories of users: student, and administrative staff (provost bodies or hall warden) users. We identified several requirements for a student user as below:

1) Each student can register an account.
2) Each student can log in to his/her account using the user credentials.
3) Each student can be identified by a unique roll number.
4) Each student can manage personal information.
5) Each student can apply for seat allocation and deallocation.
6) Each student can apply for room-related utility services.

We identified several requirements for an administrative staff user as below:

1) A user can allocate/deallocate seats for the students.
2) A user can reallocate seats for students.
3) A user can search the allocated/deallocated student's list of the hall.
4) A user can generate and collect hall fees accordingly.
5) A user can generate reports, and can publish notices and news accordingly.

Non-functional requirements are a kind of system requirements or constraint that is imposed on a system. Non-Functional requirements deal with emergent system properties such as scalability, response time, store occupancy maintainability, performance, portability, security, reliability, and many more. A list of common non-functional requirements for the system is listed below:

For domain and hosting non-functional requirements are registration of a new domain, domain panel control, purchasing hosting space from renowned service providers, hosting c-panel management, yearly payment of domain/hosting, upload speed and download speed of hosting server, monthly bandwidth of hosting, and so forth. For computer system non-functional requirements are support for all computer operating systems, modern computer browsers, computer-supported multimedia files upload and download from any computer, etc.

Non-functional requirements for Mobile devices are access from mobile devices, support for Android OS and iOS file upload and download using mobile OS, mobile screen resolution support, etc. For security the non-functional requirements are prevention of hacking and spamming, login security, registration security, prevention of malicious file upload, password recovery, etc. For ASHAMS, we identified some important non-functional requirements. These are specified shortly below: In the case of performance requirements under normal conditions, (1) the system should respond to any user request without any noticeable delay, and (2) the system should handle multiple concurrent user access and transactions.

In the case of security requirements, (1) each user should enter the system with valid credentials (user name and password), (2) the system will have options to define password policy, and (3) each user should be under a role or a set of roles. In the case of quality attributes (1) audit trial report was identified.

C. Design and Prototyping stage

The design and prototyping phase is the third stage of SDLC. Design and Prototyping are essential parts of software development. In software design, the user requirements are interpreted into the software product. In this phase, the SRS document is used as input. All of the components and security pieces of the system are also determined during the design phase.

Here, as per the SRS document, the High-level and low-level designs are prepared. Time and budget management, risk analysis, team capabilities, and project constraints are the important parameters that are also discussed in this phase. The overall system architecture is defined in this phase. Important parts of this stage are discussed below:

- Design Technique: There are four worldwide system design techniques available; these are (1) Top-down design, (2) Bottom-up design, (3) Modular design, and (4) Structured design approach.

In ASHAMS design, we use top-down design approach as it has certain lucrative features to develop software.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The Top-down design approach decomposes the whole system into smaller modules and functions. It is easy to add new features even if the development is complete.
As we know, the top-down design approach consists of smaller modules and functions; it facilitates design mistake correction at any level of design. The impact is minimal as only a single small module is affected.

- User and Hardware system requirements: Minimum specifications for the User Interfaces for ASHAMS project are (1) Front-end software: Visual Studio (VS studio 2015 or upper version); (2) Framework: the .NET Framework (4.5 or upper version); (3) Back-end software and Database software: SQL Server Management Studio (SQL Server 2019). In ASHAMS, Windows 10 64-bit OS, VS studio 2021, the .NET framework of 4.5 is used for the development purpose. Minimum specifications for the Hardware interfaces for this project are (1) Windows operating systems (Win 7,8,10 or upper); (2) HDD space (Minimum 10GB of free space); (3) Memory (8GB DDR4 RAM); (4) Processor (2.6 GHz or faster and quad-core or better recommended); (5) Display (14-inch CRT Monitor (1024X768 High Color recommended)). In addition, a 22-inch HP High-resolution Monitor, Core i9 12th generation processor of 3.70 GHz Up to 5.20 GHz clock speed, 16GB DDR 4 RAM assembled CPU is used for the development purpose.

- Use Case Diagram of ASHAMS: To illustrate the dynamic behavior, there are five available models in UML. Among those Use Case Diagram is the most popular way to describe dynamic behavior. Use case diagrams are used to describe the functionalities and requirements with the help of actors and use cases. So we can summarize that use case diagrams are comprised of several use cases, actors, and the relationships among them. An actor is an entity that interacts with the system. In software engineering, High-level functionalities are represented by the Use case. In ASHAMS, the provost (provost bodies or hall warden) and students (users) are examples of actors.

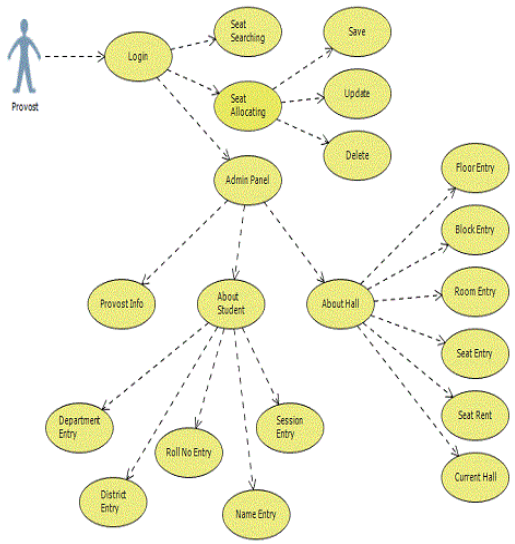

Fig. 1. Use Case diagram for Provost.

In Fig.1, we find that the participating actor is the Provost. Provost enters into the system with his/her system credentials (id and password). Here, the use case is Seat Searching, Seat Allocating, and exploring Admin Panel options. The flows of the events are Provost can save, update and delete the seats allocated. After doing the necessary work provost signs out from the system.

- ASHAMS Database design: Database design is one of the most important parts of system design. The database system organizes the data, smoothes the entire system, and increases efficiency. In addition, It controls data redundancy, enhances data integrity, data sharing and security, multiple user interfaces backup and recovery, and data abstraction. The most widespread style of a database system is the Relational Database Management System (RDBMS). In ASHAMS database design, we used stored procedures. For designing a complete database, it is required to follow four steps. These are (1) Requirement Collection and Analysis, (2) Conceptual Design, (3) Logical Design, and (4) Physical Design.

In ASHAMS database design, for the Requirement collection and analysis step we arranged several discussion sessions with the prospective database users (students and administrative staff) about the entire process. After that, we document their data requirements and the result is written as user requirements. According to the discussion sessions, we highlighted the entire process as follows:

1) There are two categories of students in a university who may be residential or not residential.
2) Only residential students are to be allocated seats by the administrative staff (provost bodies or hall warden).
3) Each student has a unique SeatCode, Name, Roll No, Phone No, Session, Department Id, Department Name, Parent's Phone No, Allotment Date, Photo, and so forth.
4) Only the residential students have to pay the seat rent.
5) Administrative staff (provost bodies or hall warden) can allocate, deallocate, and reallocate the seats upon availability.
6) Each seat has a unique SeatCode.
7) The provost has also an ID, Name, Rank, Department ID, and Department Name.
8) All the information about the student can be found in the Department.

The ASHAMS system includes accommodation management module and student management module as shown in Fig. 2. The accommodation management module includes an accommodation information module, accommodation adjustment module, accommodation import module, and audit management module. The administrator can check the accommodation information of students by inquiring about the student name, student phone number, student roll, and other conditions. In addition, the administrator can adjust student accommodation, and also check out and change applications initiated by students in the audit management.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Administrators can import accommodation information in batches through tables. The student management module consists of creating students' profiles, querying/modifying information, and deleting information.

Fig. 2. Fuction Modules of ASHAMS.

The next phase of ASHAMS database design is conceptual design. Preparing a concise summary of the requirements, and making an intricate description of the entity types, relationships, and constraints is the purpose of this step. Finding implementation details is not necessary for this phase. Moreover, Finding a report which is easier to understand and communicate with non-technical users is more important. Based on the user requirements, we identified four entity types.

Fig. 3 depicts an entity type PROVOST. ProvostId, ProvostName, and ProfessorRank are the three attributes of it. ProvostName is the composite attribute. We can specify ProvostId as the key attribute because it is specified to be unique.

PROVOST

ProvostId	ProvostName	ProfessorRank

Fig. 3. Provost Entity type.

Fig. 4 shows an entity type DEPARTMENT with attributes DepartmentId and DepartmentName. We can specify DepartmentId as the key attribute because it is specified to be unique.

DEPARTMENT

DepartmentId	DepartmentName

Fig. 4. Department Entity type.

Fig. 5 depicts an entity type SEAT with attributes BlockNo, FloorNo, RoomNo, SeatNo, and SeatCode. Here, SeatCode is the key attribute because it is specified to be unique.

SEAT

BlockNo	FloorNo	RoomNo	SeatNo	**SeatCode**

Fig. 5. Seat Entity type.

Fig. 6 shows, an entity type STUDENT with attributes StudentRoll, StudentName, Session, District, PhoneNo, AllotmentDate, ParentPhoneNo. Here, StudentRoll is specified as the key attribute because it is specified to be unique. Here, PhoneNo and ParentPhoneNo are the multivalued attributes.

STUDENT

StudentRoll	StudentName	Session	District
AllotmentDate	ParentPhoneNo	Photo	PhoneNo

Fig. 6. Student Entity type.

The relationship sets in the ASHAMS database design are listed below:

1) RESIDES: Relating Entity set Seat with the Entity set Residential with descriptive attributes SeatRent and RentDate.
2) STUDIESON: Relating Entity set Student with the Entity set Department.
3) CONTROLS: Relating Entity Set Seat with the Entity set Provost.
4) BELONGSTO: Relating Entity set Provost with the Entity set Department.

The Fig. 7 represents the Entity-Relationship diagram for the ASHAMS database. Entity types such as STUDENT, DEPARTMENT, PROVOST, and SEAT are shown in rectangular boxes. Relationship types such as RESIDES, STUDIESON, CONTROLS, and BELONGSTO are shown in diamond-shaped boxes. Attributes are shown in oval and each attribute is attached by a straight line to its entity type or relationship type. Component attributes of a composite attribute are attached to the oval representing the composite attribute as illustrated by the Name and District attribute of STUDENT. Multivalued attributes are shown in double ovals as illustrated PhoneNo attribute of STUDENT entity type, ProfessorRank attribute of PROVOST. Primary Key attributes have their name underlined. Here STUDENT entity type is specialized. Attached and Residential sub groupings are a subclass of the STUDENT entity type and this entity type is called the super class of this subclass. d is the circle stands for disjoint. In this case, d is used because we know user-defined subclasses of the specialization must be disjoint. Here in the ER diagram, we represent a double line from CONTROLS relationship to SEAT entity type and a double line from RESIDES relationship to SEAT and RESIDENTIAL entity types, which indicates a total relationship. A Provost can control all the seats. Here we can identify a double line between PROVOST and BELONGSTO

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

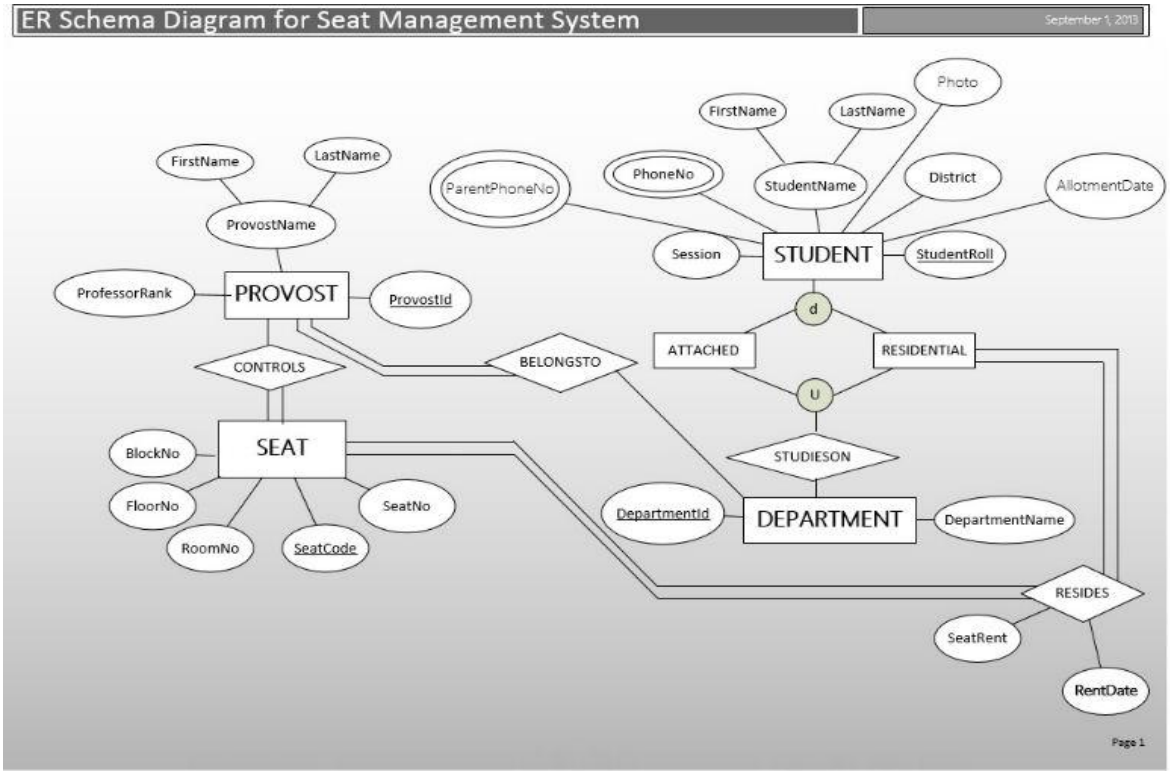

Fig. 7. ER diagram of ASHAMS database.

indicating a total relationship of PROVOST in BELONGSTO, which is each Provost, must be associated with a Department. Further, there is an arrow from BELONGSTO to DEPARTMENT indicating that each provost can have at most one associated Department. The next phase of ASHAMS database design is logical design. The purpose of this step is: (1) the conceptual schema is transferred from the high-level data model into the implementation data model and (2) It is the actual implementation of the database using a commercial DBMS. We used a High-Level Conceptual Data Model for the ASHAMS Database design. Fig. 8 represents the ER to Relational Model mapping diagram of the ASHAMS project. Here, Floors, Blocks, Rooms, Seats, Students, Address, StudentPhoneNos, SeatCategory, Resides, ParentPhoneNos, SeatRents, Departments, Provosts, ProfessorRanks, and StudentSessions are entities and each of these entities has one or more attributes. We created tables for each entity and Entity's attributes should become fields of tables with their respective data types. In addition, we declared the primary key for each table and mentioned PK.

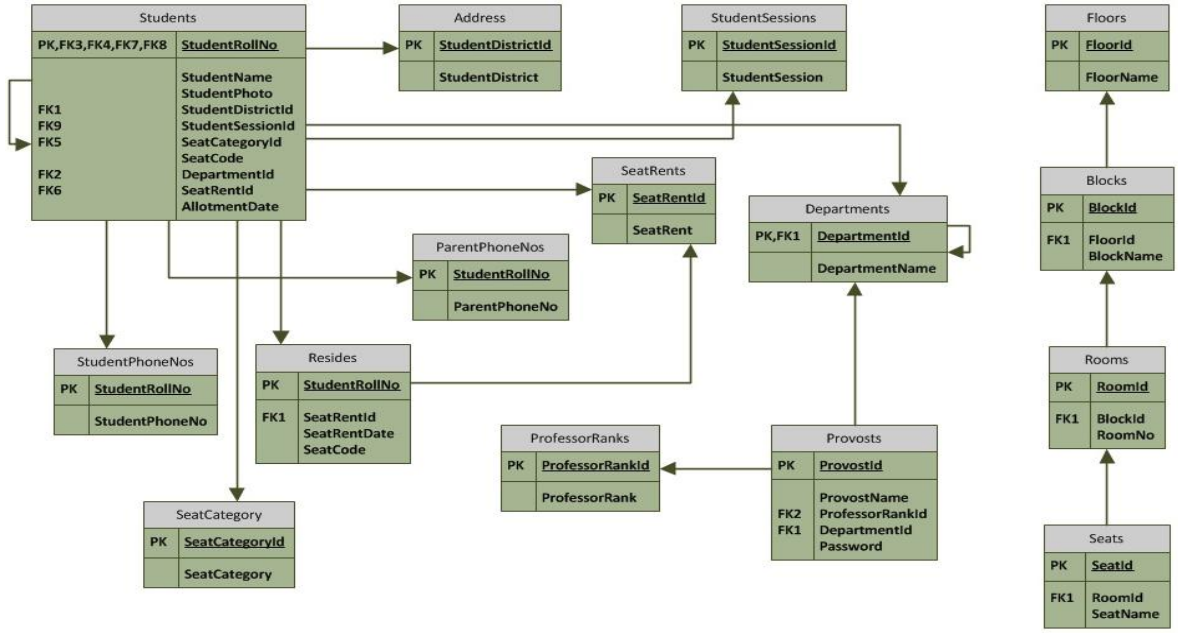

Fig. 8. ER to Relational model mapping diagram of ASHAMS.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The next phase of ASHAMS database design is Physical design. Physical database design is the process of transforming logical data models into physical data models. For relational database management system, it is very easy to convert the logical design to physical design. The database design process is also concerned with file organization of physical files. The purpose of this step is: (1) the internal storage structures, indexes, access paths, and the organization for the database files are specified. The output of our Logical design section that was mentioned earlier is used as the input for this physical section part. Now we are going to illustrate some of the tables used in the ASHAMS database below.

Users:

RONOK-PC\SQLEXP...ase - dbo.Users

Column Name	Data Type	Allow Nulls
UserName	nvarchar(20)	
UserType	nvarchar(50)	
Password	nvarchar(20)	

Fig. 9. Users table of ASHAMS database.

The above Fig. 9 illustrates the Users table and the column names are UserName, UserType, and Password. The data type of these columns is nvarchar.

Floors:

RONOK-PC\SQLEXP...ase - dbo.Blocks

FloorId	BlockId	BlockName
0	1	A
1	2	B
2	3	A
2	4	B
3	5	A
3	6	B
4	7	A
4	8	B
NULL	NULL	NULL

Fig. 10. Floors table of ASHAMS database.

Fig. 10 illustrates the Floors table and the column names are FloorId, BlockId and BlockName. The data type of these columns are nvarchar. Here, FloorId is the primary key and BlockId is the foreign key.

Departments:

RONOK-PC\SQLEX...bo.Departments

DepartmentId	DepartmentName
1	CSTE
2	PHARMACY
3	ACCE
4	FIMS
5	BBA
NULL	NULL

Fig. 11. Departments table of ASHAMS database.

Fig. 11 illustrates the Departments table and the column names are DepartmentId and DepartmentName. The data type of these columns is int and nvarchar. Here, DepartmentId is the primary key.

Seats:

RONOK-PC\SQLEXP...ase - dbo.Seats | SQLQuery12.sql - RONOK-PC\...\RO...

Column Name	Data Type	Allow Nulls
RoomId	int	
SeatId	int	
SeatName	nvarchar(20)	
BlockId	int	☑
FloorId	int	☑

Fig. 12. Seats table of ASHAMS database.

Fig. 12 illustrates the Seats table and the column names are RoomId, SeatId, SeatName, BlockId, and FloorId. The data type of these columns is int and nvarchar. Here, SeatId is the primary key.

- SDLC model: In SDLC, the most popular models are the Waterfall Model, Iterative Model, V-shaped Model, Spiral process Model, Throw away Prototype Model, Big Bang Model, Agile Model, etc. Among these, in ASHAMS we used the agile methodology. The agile model is a combination of Iterative and incremental models. In this model, the entire project is divided into small incremental builds. For ASHAMS project, the number of the developer was two. As we practiced the agile scrum methodology, we followed the questionnaire section every day. Here, agile iterations are termed sprints and each sprint lasts for two to four weeks. At the end of each sprint, the product owner verifies the product and after his approval, it is delivered to the customer. Customer feedback is taken for improvement and his suggestions and enhancement are worked on in the next sprint. Testing is done in each sprint to minimize the risk of any failures.
- Risk assessment, Cost estimation, Deliverables, Constraints, and Variance Identification: Risk is a potential threat of loss of something vulnerable. In SDLC, each phase has different dimensions and vulnerabilities of risk factors. The primary stage of risk management is to identify and understand these risks. Unwanted risks can cost a lot of money and time. Experienced developers are experts at identifying, assessing, and mitigating software risks. So, it is essential to find out the risk factors and make a proper risk mitigation plan. In ASHAMS, we found several risk factors. Table I illustrates the impact of the risks, probability of happening, and risk mitigation procedure.

Table- I: Risk Assessment for ASHAMS

RISK ASSESSMENT			
Risk Identified	**Impact (Effort, Schedule, Cost, Quality)**	**Probability of happening**	**Mitigation**
Understaning requirement	Effort, Schedule, Cost	Medium	Quick verification from subject matter expert/end users
Change on requiremets	Effort, Schedule, Cost	Low	Change management procedure

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

RISK ASSESSMENT			
Risk Identified	**Impact (Effort, Schedule, Cost, Quality)**	**Probability of happening**	**Mitigation**
Resource unavailability	Effort, Schedule, Cost	High	Resource backup plan

In software engineering, one of the hardest things is to determine the accurate cost of new software development. Every scenario is different, and there are no such thumb rules that will work for every scenario. The main target of cost estimation is estimating the amount of effort and cost of a software development project. To produce software, software companies and other organizations used manual and automated software-estimating models. For small size software, it is a general practice to use the manual software-estimating model. Enterprise software development includes automated-estimating models. In ASHAMS, we determined the type of project is a custom web-based software development. As this project involves minor changes, interaction with the clients is limited, and user interface or bug fixes are well defined so, we specified the size of the software as small size software. According to the type and size of the ASHAMS, we determined possible timeframes to develop the entire project from scratch. Table II illustrates the possible timeframes according to different features. As ASHAMS is a pilot project and we developed this project on Research and Development (R&D) based, we had to keep some buffer. We calculated and added 1 to 2 days as a buffer to each stage to keep ASHAMS on track.

Table- II: ASHAMS Project timeframe

Features	***Weeks***
Software modification (planning, requirement analysis,software design,and documentation preparation)	3-4 weeks
Web development (coding)	4-5 weeks
Software integration (testing and deployment in the hall server)	1-2 weeks
Total	8-11Weeks

After defining the type and size of the project we determined the team size. ASHAMS is a small project and we had only two teammates. So, we had to fulfill two or three different roles. One developer had to play the role of Project Manager, Business analyst, and QA tester. Another developer had to play the role of the database designer, UI/UX designer, and coder. Major variances found in ASHAMS are scope and effort variance. The common misunderstanding of all PA, lack of skilled resources, and unavailability of resources are the constraints found here. Product deliverables for ASHAMS are listed in table III. We planned to deliver our deliverables according to the demand of the customers and priority basis. We assumed there might arise changes in the template and checklists, possibilities of overlapping of PA, and new policies adaptation.

Table- III: Product deliverables for ASHAMS

Sl. No.	***Item Name***
1	Security Management
2	Static Data Management
3	Seat Allocation, deallocation, reallocation Processing
4	Half yearly information (Allocations, deallocations, Seat Rents etc.) generation for individual
5	Source code (Internal)
6	User Manual
7	Installation Manual

D. Software development (coding) stage

Coding is the longest phase of the SDLC. After finishing the design and prototyping phase of SDLC, we have to keep our eyes fixed straight ahead to the next phase, the software development (coding) phase. After receiving the design documents, the whole work is split into modules/branches and assigned to the various developers. The developers start writing code using the preferred language. Here the design draft is translated into source code, and the developers start coding according to the predefined coding guidelines. According to the industry experience, we all know that in software engineering, the maintenance cost is higher than the actual coding cost. A well-written, simple, and clean code reduces testing and maintenance costs. In addition, it will help the junior developer or new developer understand the code easily. Software developing tools (compilers, interpreters, and debuggers) are utilized to develop and implement the code. After the completion of this phase, the source code is produced accordingly, so this phase is concentrated by the developers. In ASHAMS, we tried to write clean and simple code to reduce maintenance costs. We use Microsoft Visual Studio, Microsoft Visio, and SQL Server management studio for coding purposes. As the coding progresses we tried to write test cases and perform unit testing regularly.

E. Software testing stage

According to the SDLC, once the coding is done the code is sent to the testing team. The testing team tests the proposed system's code against the requirements, and the information gathered during the requirement analysis phase. In addition, the results of existing and manual systems are compared. In this phase, functional testing (unit testing, integration testing, system testing, and acceptance testing) are conducted by the QA team. In addition, non-functional testing is also done.

In ASHAMS, according to the test plan, we performed unit testing, integration testing, and acceptance testing accordingly. The unit testing is conducted first before coming to the system integration test. In unit testing, we tested the individual functions according to the logic and match those accordingly. In Integration testing apart from taking module by module, we take the interface as a whole. User Acceptance Test conducted to validate system as per requirement. We incorporated and tested the server and client module options during this phase. Before the implementation, the ASHAMS is tested by the users (provost bodies, students and hall warden).

F. Implementation and Integration stage

One of the most important parts of the SDLC is the implementation and integration phase. The final deployment process starts after the successful conclusion of the testing period and confirmation of zero bugs or errors. As we all know software development is a continuous process, the system may not fulfill all the requirements or may fail in a specific situation. In ASHAMS, after being passed the UAT (User Acceptance Testing) we deployed the system on the test server. This first version of ASHAMS was considered the beta version. After successful beta testing, the system was deployed on the live server (Abdus Salam hall server).

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

G. Operations and Maintenance stage

After implementation, the users start using the system and there often occur three important scenarios as below:

1) Bug fixing: There is a possibility of producing bugs that should be addressed immediately.
2) Enhancement: For any kinds of modifications and enhancements required according to the user's new requirement should be addressed by the developers.
3) Upgrade version: After the regular interval, the deployed application needs to be upgraded to a newer version.

IV. RESULTS AND DISCUSSIONS

Fig. 13 and 14 shows two different login pages. Fig. 13 shows the Administrator Login page (provost bodies or hall warden) module, and Fig. 14 shows the Student Login page module. The system captured student details and other necessary information from the hall office that is used to create the students' accounts. The system will verify and validate all user input accordingly. The Administrator login module consists of three parts: provost name, provost rank, and password.

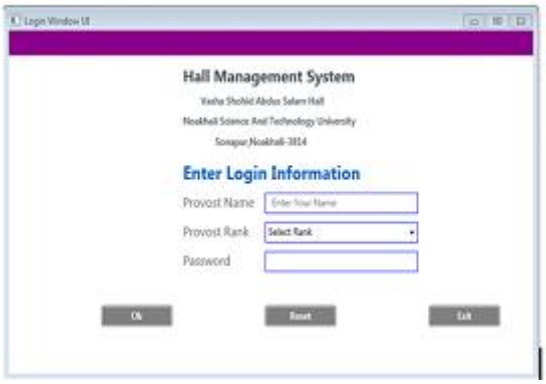

Fig. 13. Login window (Administrator) of ASHAMS.

The Student login module consists of four parts: Batch, program, roll no, and password. In both cases, users have to use appropriate credentials that were stored in the database earlier. The administrator can log into his account. If the user name and password are incorrect, an incorrect user name or password prompt will appear.

Fig. 14. Login window (Student) of ASHAMS.

After successful login, we will get a window like the below (Fig.15). This window will show who is currently entered in the system and in the top menu bar, there is a menu for seat allocation management.

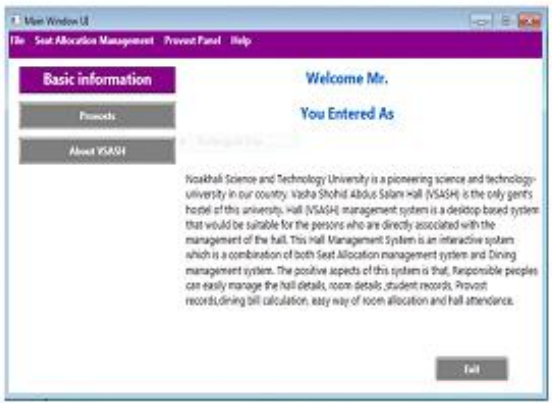

Fig. 15. Main window of ASHAMS.

The Fig. 16 depicts the Hall information Entry and Update window. Using this window, the System Administrator (Provost Bodies or Hall Warden) can add, update and delete the floors, blocks, rooms, and seats information accordingly. The system administrator can easily keep track of the hall information quickly.

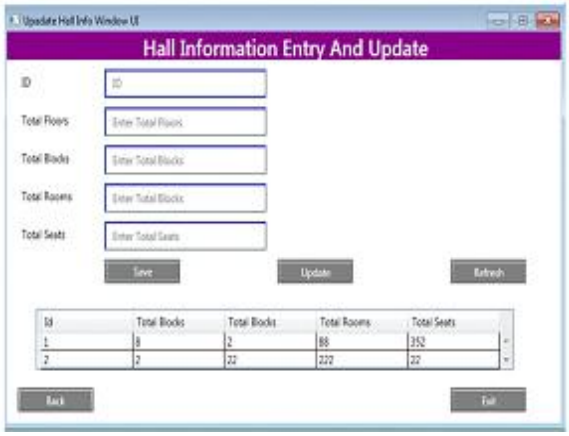

Fig. 16. Hall information entry and update window of ASHAMS.

Then The new allotment window is shown in Fig. 17. From this page, the System Administrator can easily make a new seat allotment filling up all the associated fields accordingly.

Fig. 17. New Allotment window of ASHAMS.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Fig. 18 depicts the students' details at a glance. It is often required to know the student details (current status of a specific student) at a glance. Here the administrator can search for a user based on Phone Number or the Seat Code.

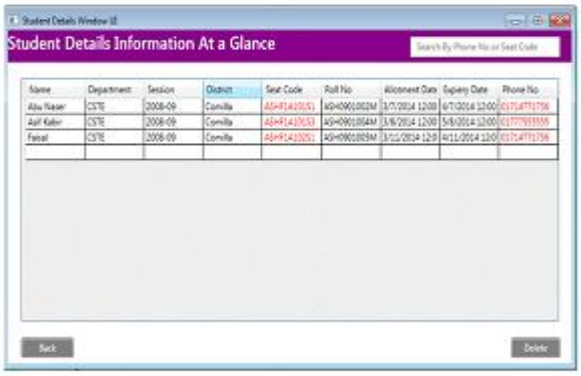

Fig. 18. Student details at a glance window of ASHAMS.

Fig. 19 and Fig. 20 show the BSASH at a glance and the developer window at a glance. A pictorial representation of the current stucture is shown in the figure. Here, the BSASH consists of two blocks, four floors, hall dining, and mosque. Fig. 20 depicts the developer's information and system details information.

Fig. 19. BSASH at a glance window of ASHAMS.

Fig. 20. About Developer window of ASHAMS.

V. CONCLUSION

The proposed ASHAMS is designed to automate the overall tedious accommodation management system. This system provides convenient and efficient system functions for both students and administrations. Our main focus was to develop a fast responding, error-free, cost-effective, and redundant data-free system. Unnecessary administrative tasks and paperwork will be reduced through the system. We hope this system can minimize the shortcomings of the existing manual and traditional accommodation management systems. Based on the outcome and fact findings, we hope to expand the service scope better and integrate other lucrative management systems like Hall Library Management System and Meal Management System with this system. In addition, reporting system for accommodation management, library management, and dining transactions will be added later. We hope this system will be helpful for NSTU and bring one step forward for the sustainable development of the socio-economic infrastructure and our dream to make our country a digital Bangladesh.

REFERENCES

1. Peng, Z., Liu, T., & Mai, L. (2020, August). Design and Implementation of Dormitory Management System based on SSM framework. In 2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS) (pp. 321-325). [CrossRef]
2. Ayanlowo, K., Shoewu, O., Olatinwo, S. O., Omitola, O. O., & Babalola, D. D. (2014). Development of an automated hostel facility management system. Journal of Science and Engineering, 5(1), 01-10.
3. Zhang, X., Hu, Y., Lu, Y., & Gu, J. (2011, August). University dormitory management system based on agile development architecture. In 2011 International Conference on Management and Service Science (pp. 1-4). IEEE. [CrossRef]
4. Lubis, M., Fauzi, R., Lubis, A. R., & Fauzi, R. (2018, August). A Case Study of Universities Dormitory Residence Management System (DRMS) in Indonesia. In 2018 6th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-6). IEEE. [CrossRef]
5. Zhang, X., Hu, Y., Lu, Y., & Gu, J. (2011, August). University dormitory management system based on agile development architecture. In *2011 International Conference on Management and Service Science* (pp. 1-4). IEEE. [CrossRef]
6. Kumar, G., & Bhatia, P. K. (2012). Impact of agile methodology on software development process. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, *2*(4), 46-50.
7. Williams, L. (2010). Agile software development methodologies and practices. In *Advances in computers* (Vol. 80, pp. 1-44). Elsevier. [CrossRef]
8. Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). Software development life cycle AGILE vs traditional approaches. In *International Conference on Information and Network Technology* (Vol. 37, No. 1, pp. 162-167).
9. Bhuvaneswari, T., & Prabaharan, S. (2013). A survey on software development life cycle models. *International Journal of Computer Science and Mobile Computing*, *2*(5), 262-267.
10. Shylesh, S. (2017). A study of software development life cycle process models. In *National Conference on Reinventing Opportunities in Management, IT, and Social Sciences* (pp. 534-541).
11. Zhang, Z. (2010, April). Design and Implementation of Accommodation Management Information System Based on SQL. In 2010 International Conference on Machine Vision and Human-machine Interface (pp. 457-460). IEEE. [CrossRef]
12. Garcia, D. E. N. (2014). Accommodation management system using mobile devices (Doctoral dissertation).
13. Maren Bell, Y., Ruiz Obret, D., & Pérez García, J. C. (2021). Accommodation module for the university accommodation process, at the Universidad de Oriente.
14. Raj, G., Singh, D., & Bansal, A. (2014, September). Analysis for security implementation in SDLC. In *2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence)* (pp. 221-226). IEEE. [CrossRef]

AUTHORS PROFILE

Ronok Bhowmik, received the B.Sc. Engg. in Computer Science and Telecommunication Engineering from the Noakhali Science and Technology University (NSTU), Bangladesh in 2012. He completed his M.Sc. in Computer Science from Jahangirnagar University (JU), Bangladesh in 2016. Currently, he is working as a Programmer in the Cyber center at NSTU. His main research interest includes the Internet of Things (IoT), Software Engineering, and Machine Learning.

Md. Hasnat Riaz, is working as an Assistant Professor in the Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University. His research interest includes Cloud Computing, Mobile Cloud Computing, IoT, Data Science, and Machine Learning.