**smart home management**

**4. Recommended Solution: Describe the solution that you will use to solve the problems identified in your business case.**

The recommended solution entails the development of a comprehensive smart home application that integrates hardware, software, processes, data, and people. The core components of this solution include smart sensors and cameras for security purposes, home automation tools for appliance control and energy management, and assistive tech for aiding handicapped and old-age individuals. These components are orchestrated through a mobile application and web interface which provide remote access and control to users. To ensure interoperability with various devices, our solution will integrate with existing smart home platforms and protocols. In terms of software engineering, we plan to follow an Agile development methodology, specifically Scrum, to ensure iterative progress and adaptability in our development process. This approach allows us to incorporate changes quickly and efficiently while maintaining focus on delivering a quality product. For data storage and analytics, we propose a centralized database system that captures user preferences, security logs, and energy consumption data. This data can then be analyzed using built-in analytics tools to generate actionable insights and recommendations for the users. Quality assurance is another critical aspect of our solution. Through continuous testing, we aim to maintain a high level of performance reliability in our smart home application. Lastly, recognizing the importance of privacy and data protection, our solution complies with all relevant regulations to safeguard user's information (Genever, 2021).

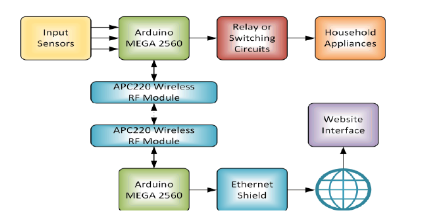
**4.1. Apply a proper Design Pattern to the recommended solution.**

The recommended solution for the Smart Home application can be realized using a combination of design patterns. The Model-View-Controller (MVC) pattern is initially applied to separate the system's concerns, where the 'Model' represents the data and business logic, the 'View' handles the user interface and presentation, and the 'Controller' coordinates interactions between the Model and View. Further, the Observer pattern is used in tandem with MVC. In this smart home context, the Observer pattern allows various parts of the system (Observers), such as security sensors, assistive devices, or energy management components, to react when changes occur in some monitored object (Subject). For instance, if there's an unexpected movement detected by a sensor (Subject), it notifies relevant parts of the system like alert systems or mobile notifications (Observers). Additionally, the Factory Method Pattern is proposed for creating different types of automated devices without exposing creation logic to client code. This pattern facilitates scalability and flexibility as new device types can be added without changing existing code. For example, if a new type of assistive device is introduced, the Factory Method Pattern ensures smooth integration into the existing system. These combined design patterns provide an efficient, scalable, and flexible architecture for the smart home application (Rosslin John Robles, & Kim, 2010).

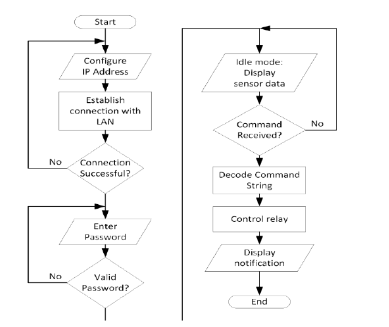
**4.2. Describe the solution in terms of several perspectives like What, How, Where, Who, When and Why factors.**

The solution to the smart home application is comprehensive, addressing various factors such as What, How, Where, Who, When and Why. What: The solution includes a combination of hardware and software components to create a smart home environment. Smart sensors, cameras, home automation devices, assistive technology, mobile applications and web interfaces are all part of this ecosystem. How: Agile development methodology (Scrum) is used for progressive and iterative product development. Continuous testing ensures reliable performance. A centralized database stores user preferences, security logs, and energy consumption data, while analytics tools generate insights for users. Where: The solution can be implemented in any residential setting with an internet connection. It is integrated with existing smart home platforms and protocols for seamless operation. Who: This system is especially beneficial to handicapped and senior citizens who need assistance managing their daily activities. Additionally, it can provide convenience for any homeowner interested in smart home technology. When: The system operates 24/7, providing round-the-clock security monitoring, energy management, and personalized support for individuals with special needs. Why: The solution addresses the challenges faced by individuals who struggle to manage their routine activities due to physical limitations or age-related issues. It also enhances the security of the home environment and aids in efficient energy management. This solution abides by relevant data protection laws and privacy regulations to ensure user information confidentiality (Kim et al, 2020).

**4.3. Create a working prototype to model your solution. You can use Justinmind/draw. Software for prototyping.**



The prototype for the smart home application was created using Justinmind, a comprehensive software modeling tool. The mobile control system was used as the foundation for the evolution process in the design that was proposed. A mini web server that is based on Ethernet is included in the SHCS configuration along with a website platform. It showcased the user interface and interaction flow of the application, providing an immersive and interactive experience to stakeholders during the design phase of the project. In terms of user interface, the prototype featured a simple and intuitive dashboard that displayed real-time information about security status, energy consumption levels, and the status of connected devices in the home. The interaction flow was designed to be user-friendly, enabling users to easily navigate through different sections such as device control, settings, reports, and alerts. Moreover, features like voice command assistance were integrated into the prototype to support handicapped or old-age users. Remote access capability was also demonstrated in the prototype, allowing users to control their home appliances and security systems from anywhere via the mobile application or web interface. Furthermore, the prototyping process allowed us to identify potential challenges in the system integration and interaction flow. This enabled us to address these issues during the early stages of development, saving time and resources in the later stages. The prototype served not only as a visual representation of our solution but also as a valuable tool for continuous improvement and innovation (Teddy Surya Gunawan et al, 2017).



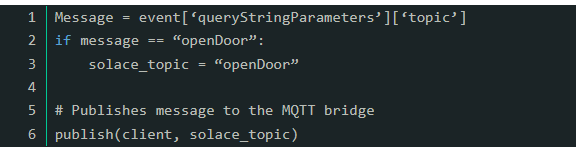
To begin, users will need to configure the SHCS's IP address and then enter that address into the web browser's address bar. The SHCS prototype is connected to the network router at the university, which has a dynamic and private IP address. This allows the prototype to function. Next, the user will be prompted to enter the password in order to go to the main page. The website is maintained in an inactive state and reloaded once per second, with the frequency being programmable, so that it can be kept up to date with the most recent sensor information. Last but not least, if the user presses a command key, a command string will be decoded. Following the microcontroller's interpretation of the command, it will either produce a HIGH or LOW output to the relay circuit. The relay circuit makes it possible for a low-voltage Arduino to exercise control over high-voltage household appliances (Teddy Surya Gunawan et al, 2017).

**4.4. Describe the type of developed software achieved: Data, process, technology. Use models and xml code to support your discussion**



Our design implementation included interactions between

* AWS API Gateway
* A Lambda function on AWS
* Solace PubSub+ Event Broker
* Arduino



The user initiates the workflow by clicking a button within the online application to begin the process. Because you clicked the button, this interaction will cause a GET request to be sent over the API gateway.An AWS lambda function that was coded in Python was activated when the API gateway was used. The message body that was included in the GET request served as the sole criteria for determining which Solace topic this function will publish to. For instance, if "openDoor" was the message that was received from the GET body request, then the Solace PubSub+ Event Broker would publish the openDoor topic. When we say this, we are essentially claiming that the web application is the "Publisher" of events (Loshin et al, 2021).

The developed software for the smart home application is a multi-tiered system integrating data, processes, and technology. Data: The software utilizes several types of data ranging from user preferences to security logs and energy consumption figures. A centralized database stores all these diverse data sets which are collected through IoT devices installed in the home. This data is analyzed by the analytics module that generates insights and recommendations for improving energy efficiency. Processes: The software follows an event-driven process model where actions are initiated based on specific events or triggers (e.g., motion detection, temperature changes). The Scrum methodology was employed to facilitate iterative development and continuous testing, ensuring reliable performance. Technology: The technology stack includes smart sensors, cameras, home automation devices, a mobile application, and a web interface. The software integrates with existing smart home platforms via standard protocols like Zigbee and Z-Wave. A simple XML code snippet representing a sensor's data might look something like this: ```xml Motion Lounge Active 2022-02-11T18:25Z ``` This signifies a motion sensor located in the lounge that was last triggered at the specified time. Such use of XML helps maintain interoperability between different components of the smart home ecosystem (Loshin et al, 2021).

**5. Testing Plan. Describe your plan to complete testing of the recommended solution including:**

The testing plan for the proposed smart home application will be an integrated part of the Agile development process, with continuous testing and quality assurance activities woven into each sprint. At the start, unit testing will be conducted to validate each piece of code in isolation. As individual software components are developed, integration testing will follow to ensure these components work together as expected. Once the initial version of the application is ready, system testing will be implemented to test the app as a whole. This phase will also include conducting performance, security, and usability tests. The performance testing will check the system’s response time and stability under varying loads. Security testing will involve vulnerability scanning and penetration testing to identify potential threats and weaknesses. Usability testing will ensure the user interface is intuitive and user-friendly, particularly for handicapped and old-age individuals. In parallel, acceptance testing will be carried out by a group of selected end-users who fit the target user profile. They will use the application in real-world scenarios to verify if it meets their needs and expectations. Feedback from this phase will be crucial in refining and improving the application before its official launch. Throughout this process, automated testing tools and frameworks will play a pivotal role in boosting efficiency and accuracy (Gülay Başol et al, 2017).

**5.1. Testing Stages and Testing Techniques**

The testing stages and techniques utilized during the development of the smart home application were crucial in ensuring the software's functionality, usability, and reliability. The initial stage involved Unit Testing, where individual functions or procedures within the software were tested to verify their correctness. This was followed by Integration Testing, where all modules were combined and tested as a group to ensure they functioned together seamlessly. Subsequently, System Testing was carried out to validate the entire system’s compliance with the specified requirements. It examined both functional and non-functional aspects of the software such as responsiveness, performance, and security features. User Acceptance Testing (UAT) was then conducted, which essentially involved real users testing the software in real-life scenarios to ensure it met their needs and expectations. In parallel to these stages, various testing techniques were employed. Automated Testing was used extensively for regression testing to quickly identify new bugs in existing functional components after modifications. Manual Testing was implemented for exploratory testing purposes to investigate less predictable elements of user interaction. Load Testing was performed to evaluate how well the software could handle multiple users simultaneously, while Security Testing checked for potential vulnerabilities in the system. Lastly, Accessibility Testing ensured that the assistive technology was user-friendly and effective for elderly and handicapped individuals.

1. Know what you're testing and why. The first step is to know what you want to find out about your product through testing. Plus, explain how you plan to share your results and use them in the development process (Jordi Mongay Batalla, & Franciszek Gonciarz, 2018).

This helps you figure out what kind of sample will be most helpful for your design direction. These are different ways to test your product's prototype at different stages:

* When you are making a new product (MVP), you should test a product sample. This could be something brand new, or it could be a product you plan to sell in addition to the ones you already have.
* Testing a product redesign is a way to get ideas for a big redesign. You have a product, but you want to change it in a big way to make it better. For example, it can help show what changes need to be made to the information design to make it easy to understand and use (Jordi Mongay Batalla, & Franciszek Gonciarz, 2018).
* Feature prototype testing is a way to check out new features before they are built. For example, beta testing is a type of feature prototype testing that lets you get feedback from a certain group of your users.
* In-product experiments testing is a way to test current products on a smaller scale and see if an idea works through in-app messaging. You want to get better at a certain skill and get as much feedback as possible. For this, you can use the Experiment Canvas, which was made by the WRKSHP Tools described above. It will help you get the right people talking at the right time about the right experiment.

2. Decide who you want to take your test.

* Next, you need to decide who will try your prototype. These people are called "test participants." You want the people you test with to be like the people you want to use your product.
* For example, if your product hasn't come out yet, you should look for ICPs (ideal client profiles) of people who would be interested in it. In our example of a social app for business events, our ICPs could be workers who often meet new people while out and about. They are the ones for whom the app was made, so they are the best people to talk to for useful comments (Jordi Mongay Batalla, & Franciszek Gonciarz, 2018).

3. Make a test model #

Depending on where you are in testing and development, you will make a different type of prototype. You should think about two main types of prototyping:

* Early on in the process of making a new product, low-fidelity samples are used to test ideas. They are often simple wireframes or paper prototypes that let you get early feedback and insights. Low-fidelity prototypes can help you with the basics, like figuring out if your plan works or if users can find basic information.
* In the last steps of your design process, you test with high-fidelity prototypes. At this point, you should feel pretty good about your design, and trying the prototype should only show you small problems with how it works. For example, your high-fidelity prototype should help you learn more about your copy, user flows, and any other final-stage factors (Jordi Mongay Batalla, & Franciszek Gonciarz, 2018).

4. Choose the method for checking usability.

Your usability testing method is how you will gather feedback from users. You should think about a number of customer testing methods, such as:

* Moderated vs. unmoderated: During a moderated test, like guerilla testing, someone from your team will be there to help guide you and give you directions. In an unmoderated test, like open card sorting, the people taking the test will be given directions and tasks to do ahead of time, and then they will test the prototype on their own.
* Remote vs. in-person: This one is pretty easy to understand. When the customer and the researcher are in different places, this is called remote usability testing. In-person usability testing needs users to be there for the test. This is usually done in an office, a testing lab to reduce bias, or a user's home to give them the kind of everyday distractions they would normally have.
* Quantitative vs. Qualitative: You get numbers from quantitative insights, like those you get from A/B testing. For instance, it could be how long it takes users to finish a certain job. Qualitative data, like 5-second tests, gives you verbal or written feedback, like what people say about your prototype when they try it (Jordi Mongay Batalla, & Franciszek Gonciarz, 2018).

5. Choose the tools you will use.

* People say that a worker is only as good as the things he or she uses. The tools you use to test your prototypes with users can make or break the process.
* You need a tool to help you build your prototype first. Low-fidelity prototypes can be made by hand and on paper, but for high-fidelity prototypes, you'll need something with a bit more detail.

**5.2. Samples of how testing can be achieved.**

Testing for the smart home application can be achieved through various methods to ensure its reliability and efficiency. One of the primary testing approaches is Unit Testing, where individual components of the software are tested in isolation. This includes testing each function, interface, and module to verify it works as intended. Another approach is Integration Testing which focuses on checking the interaction between different modules. This test would identify any discrepancies in data exchange between sensors, home automation devices, mobile application, and web interface. Functional Testing can also be conducted to determine if the system meets specified requirements. For example, a scenario could be created where the user tries to remotely control an appliance through the mobile app; the system should carry out this task successfully. Performance Testing would measure how well the system performs under load, especially during peak usage times. This would provide insights into potential bottlenecks and areas that need optimization. Security Testing is crucial in such applications to protect user data and prevent unauthorized access. It involves vulnerability scanning, security audits, and penetration testing. Finally, Usability Testing can be carried out with real users to understand their experience with the system's user interface and interaction flow. Their feedback can be invaluable in refining and improving the system design. All these tests contribute to building a reliable and efficient smart home application (Kim, 2010).

References

Genever, H. (2021, August 3). *10 Step Process for Effective Business Problem Solving*. LivePlan Blog. <https://www.liveplan.com/blog/the-best-ways-to-approach-problem-solving-in-business/>

Gülay Başol, Rüştü Güntürkün, & Erkan Başol. (2017, June 22). *Smart Home Design and Application*. ResearchGate; unknown. <https://www.researchgate.net/publication/317829748_Smart_Home_Design_and_Application>

Jordi Mongay Batalla, & Franciszek Gonciarz. (2018). *Deployment of smart home management system at the edge: mechanisms and protocols*. *31*(5), 1301–1315. <https://doi.org/10.1007/s00521-018-3545-7>

Kim, M.-N., Cho, M.-H., & Jun, H.-J. (2020). *Developing Design Solutions for Smart Homes Through User-Centered Scenarios*. *11*. <https://doi.org/10.3389/fpsyg.2020.00335>

Kim, T. (2010). *Applications, Systems and Methods in Smart Home Technology: A Review*. Academia.edu. <https://www.academia.edu/6341513/Applications_Systems_and_Methods_in_Smart_Home_Technology_A_Review>

Loshin, P., Linthicum, D., & Giza, M. (2021). *XML (Extensible Markup Language)*. WhatIs.com; TechTarget. <https://www.techtarget.com/whatis/definition/XML-Extensible-Markup-Language>

Rosslin John Robles, & Kim, T.-H. (2010). *Applications, Systems and Methods in Smart Home Technology: A Review*. ResearchGate; unknown. <https://www.researchgate.net/publication/242630611_Applications_Systems_and_Methods_in_Smart_Home_Technology_A_Review>

Teddy Surya Gunawan, Intan Rahmithul Husna Yaldi, Kartiwi, M., & Anis Nurashikin Nordin. (2017, July). *Prototype Design of Smart Home System using Internet of Things*. ResearchGate; Institute of Advanced Engineering and Science. <https://www.researchgate.net/publication/319345708_Prototype_Design_of_Smart_Home_System_using_Internet_of_Things>

‌

‌

‌