To design and implement an anytime electricity bill payment controller, you would need to consider several components and functionalities. Here's a high-level overview of the design and implementation process:

1. Requirements Gathering: Understand the requirements of the electricity bill payment system. This includes identifying key features, such as user registration, bill retrieval, payment processing, notifications, and security considerations.

2. Database Design: Determine the database schema to store user information, billing details, and payment history. Consider using a relational database like MySQL or a NoSQL database like MongoDB, depending on the specific requirements.

3. User Registration and Authentication: Implement a user registration system to allow customers to create accounts. Use authentication mechanisms like username/password or third-party authentication providers (e.g., Google, Facebook) to secure user access.

4. Bill Retrieval: Integrate with the electricity service provider's system or APIs to retrieve the latest bill details for a given user. This could involve fetching bill amount, due date, and other relevant information.

5. Payment Gateway Integration: Integrate with a payment gateway (e.g., Stripe, PayPal) to handle payment processing. Set up secure payment transactions and handle responses from the payment gateway, including success, failure, and cancellation scenarios.

6. Notifications: Implement a notification system to send reminders to users regarding upcoming bill due dates, successful payments, and any other relevant updates. This can be done through email, SMS, or push notifications, depending on the user's preferences.

7. Billing History and Reports: Provide a feature for users to view their payment history and generate reports, such as monthly or yearly summaries. This allows customers to track their usage and expenditure over time.

8. Security Measures: Implement appropriate security measures to protect user data and transactions. This includes secure communication protocols (HTTPS), encryption of sensitive information, and following best practices for data storage and handling.

9. User Interface: Design and develop a user-friendly interface for customers to interact with the payment controller. Consider responsive web design principles to ensure accessibility across various devices.

10. Testing and Deployment: Conduct thorough testing to ensure the system functions correctly, including unit tests, integration tests, and user acceptance testing. Once testing is complete, deploy the application to a production environment, considering scalability and performance requirements.

11. Ongoing Maintenance: Continuously monitor the application, perform regular updates and patches to address any security vulnerabilities or bugs. Also, gather user feedback to improve the user experience and add new features as needed.

Remember that this is a high-level overview, and the specific implementation details may vary based on your requirements and the technologies you choose to use.