

SYSTEM MAINTENANCE

DCARD APPLICATION



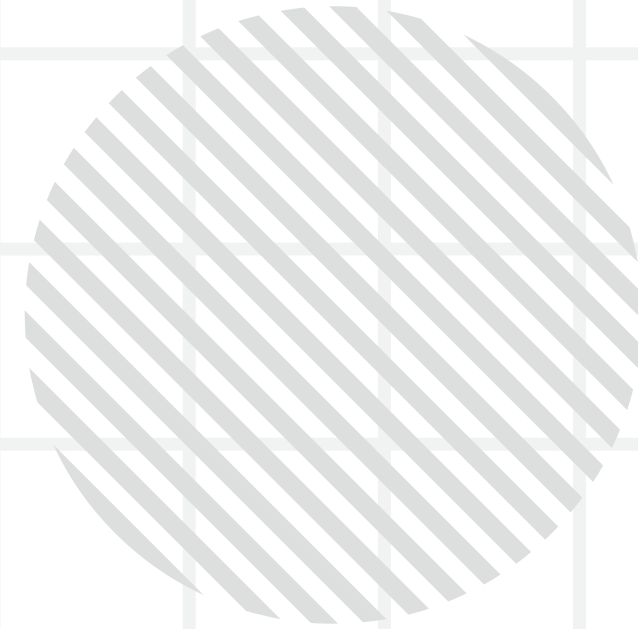
Done by:

Alex Hu Qiao Feng (23SMD05898)

Natalie Koa Hao Yee (23SMD01336)

Tan Shieh Ling (23SMD00488)

Michelle Chin Koh Ying (23SMD05432)



1.0 INTRODUCTION



DCARD



A popular social networking platform with over 1 000,000 + downloads, available on multiple platforms such as Android, IOS, Windows, and MacOS.

FUNCTIONS OF DCARD

LOGIN & REGISTER

Where users can create a new account or log in the account user created.

VALIDATION

to validate user account and to validate user is from a specific institution.

SEARCH

To help user find specific content, discussions, and other users quickly and efficiently

MATCHMAKING FEATURES

Help users to find potential friends or partners based on shared interest and preferences

POST

To post content either openly or anonymously



FAULTS & ERRORS

PERFORMANCE ISSUES

platform experiences
occasional performance
slowdowns or
downtime during peak
hours.

USER INTERFACE

Navigation is not
direct, it is buried
under multiple
menus and are
difficult to locate

BUGS

Glitches like
unresponsive
buttons,
misaligned
elements, and
overlapping text.

LOGIN & VALIDATION ISSUES

System fails to let
user login and
validate user.

CHANGES

OPTIMIZE PERFORMANCE & SCALBILITY

by upgrading server infrastructure, implement load balancing techniques, and optimize database queries to ensure platform can handle increased traffic efficiently

ADDRESS STABILITY ISSUE

conduct thorough testing and debugging to identify and fix underlying causes. implementing robust error handle and recover mechanisms will help prevent data loss and ensure application can recover from unexpected issue

FIXING BUGS

By regularly updating the app to fix the issues, along with implementing quality assurance process to help ensure a smoother user experience

FIXING LOGIN & VALIDATION

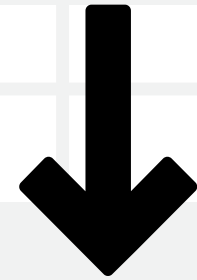
Commit regular updates, expand database of recognized university email domains, validation system error-handling, improving the reliability of email confirmation process.



2.0 FACTORS THAT INFLUENCE SOFTWARE MAINTENANCE

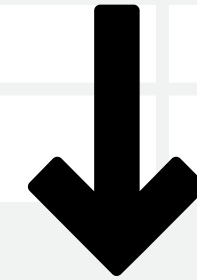


FACTOR THAT INFLUENCE SOFTWARE MAINTENANCE



STAFF

- Knowledge & Experience
- Focus



SYSTEM

- Size
- Complexity
- Life expectancy

STAFF

KNOWLEDGE & EXPERIENCE

- good knowledge and experience of the architecture, codebase and business logic.
- lack of experience in the system will slow down the maintenance greatly.

FOCUS

- distracted by multiple projects
- deteriorating system performance or user dissatisfaction

SYSTEM

SIZE

- 11 MB on Android and 88 MB on IOS
- a large user database with millions of users and a significant amount of daily active users.

COMPLEXITY

- login function, posting function, and comment function.
- developers need to adhere to modular design principles and minimize interdependencies between different parts of the system

LIFE EXPECTANCY

- established in 2011, which means that it was a relatively mature system
- constant maintenance to update its software stack, refactor old code, and keep up with advancements in the social media ecosystem



3.0 SOFTWARE MAINTENANCE FRAMEWORK



3.0 SOFTWARE MAINTENANCE FRAMEWORK

USERS

- **Primary Users:** Initially university students, discussing academics, relationships, and campus life.
- **User Expansion:** Now includes the general public, broadening discussion topics.
- **Interest Groups:** Users engage in forums based on hobbies and shared experiences.
- **Anonymous Interaction:** Encourages open, honest discussions.

3.0 SOFTWARE MAINTENANCE FRAMEWORK

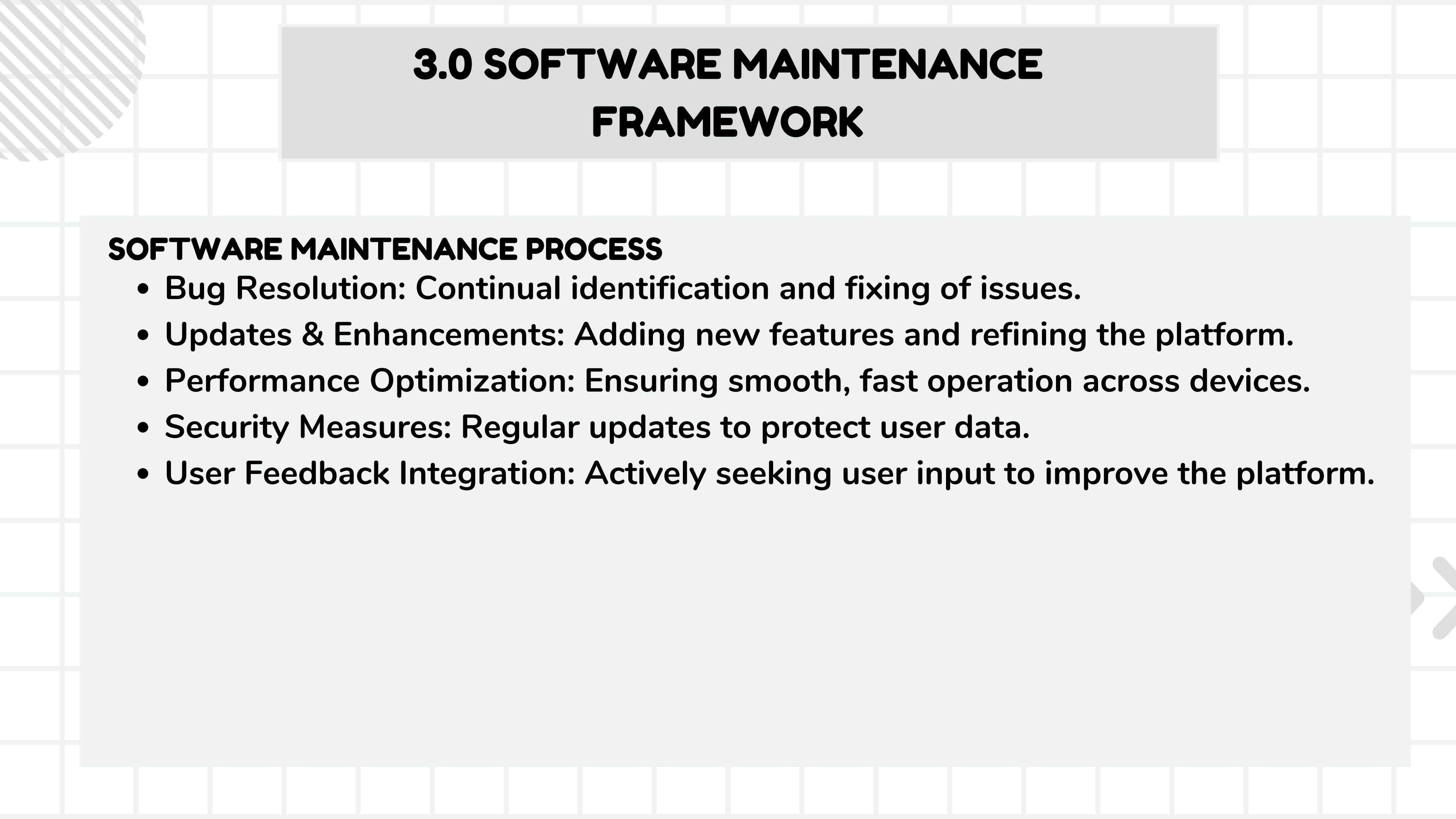
ENVIRONMENTS

- **Mobile Platforms:** Apps available for Android and iOS; maintenance required for OS updates and device compatibility.
- **Web Platforms:** Accessible on various browsers, with a focus on cross-browser compatibility and performance optimization.
- **Server Environment:** Uptime, security, and scalability to support growing traffic and data.



3.0 SOFTWARE MAINTENANCE FRAMEWORK


SOFTWARE MAINTENANCE PROCESS

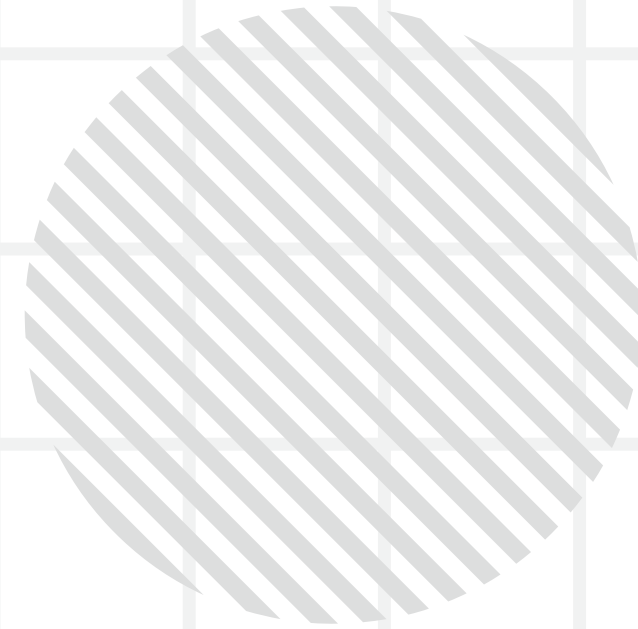
- **Bug Resolution:** Continual identification and fixing of issues.
 - **Updates & Enhancements:** Adding new features and refining the platform.
 - **Performance Optimization:** Ensuring smooth, fast operation across devices.
 - **Security Measures:** Regular updates to protect user data.
 - **User Feedback Integration:** Actively seeking user input to improve the platform.
- 



3.0 SOFTWARE MAINTENANCE FRAMEWORK

PRODUCT FEATURES

- **User Profiles:** Customizable profiles with pictures and personal info.
 - **Instant Messaging:** Enabling private conversations between users.
 - **Multimedia Sharing:** Sharing images, videos, and other content.
 - **Event Organization:** Tools to plan and promote events for community engagement.
- 



4.0 MAINTENANCE PLANNING



4.1 INTRODUCTION

```
graph TD; A[4.1 INTRODUCTION] --> B[SYSTEM DESCRIPTION]; A --> C[SUPPORT ORGANIZATION];
```

SYSTEM DESCRIPTION

Dcard is a popular social networking platform, initially designed for university students and later expanded to the general public.

SUPPORT ORGANIZATION

Dcard Corporation is responsible for the maintenance, with protocols established between the customer and the maintainer to ensure smooth operations and support.

4.2 MAINTENANCE CONCEPT

```
graph TD; A[4.2 MAINTENANCE CONCEPT] --- B[SUPPORT LEVELS]; A --- C[SUPPORT PERIOD]; A --- D[PROCESS TAILORING];
```

SUPPORT LEVELS

Define the levels of support required to keep Dcard operational across different platforms.

SUPPORT PERIOD

Establish the duration for which maintenance support will be provided, including any planned updates or enhancements.

PROCESS TAILORING

Customize the maintenance process to address the specific needs of Dcard, ensuring that it remains relevant and functional.




4.3 ORGANIZATION AND MAINTENANCE ACTIVITIES

⚙️ PRE-DELIVERY:


- Set up the necessary infrastructure and training processes.
- Develop and document the maintenance process.

⚙️ POST-DELIVERY:

- Implement procedures for problem analysis, modifications, review, and acceptance.
 - Manage the migration and eventual retirement of outdated components.
 - Provide support through help desk services and ongoing training for both maintainers and users.
 - Continuously improve the maintenance process based on feedback and performance metrics.
- 



4.4 RESOURCES: MAINTAINING DCARD REQUIRES SPECIFIC RESOURCES

- **Personnel:** A skilled team of developers, testers, and support staff to handle ongoing maintenance.
 - **Software & Hardware:** The necessary software tools and hardware infrastructure to support the platform's functionality across various environments.
 - **Facilities:** Virtual or physical spaces required for development, testing, and maintenance activities.
 - **Documentation:** Comprehensive documentation, including maintenance manuals, test plans, and user manuals, is crucial for efficient maintenance.
 - **Data & Other Resources:** Additional resources, such as backup systems and monitoring tools, to ensure the application's smooth operation.
- 

4.5 PROCESS: THE MAINTENANCE PROCESS FOR DCARD

PROCESS OVERVIEW

A high-level overview of the maintenance activities without delving into granular details.

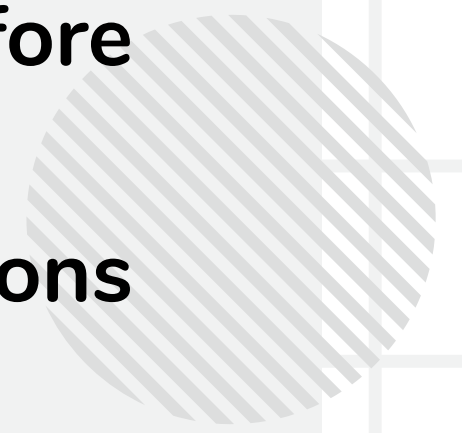
TAILORING

Adjust the process to meet Dcard's specific requirements.



4.5 PROCESS: THE MAINTENANCE PROCESS FOR DCARD

⚙ EXECUTION STEPS:

- **Identification:** Collect detailed information about issues from logs, user reports, and monitoring tools.
 - **Prioritization:** Rank issues based on their impact on Dcard's user experience and functionality.
 - **Planning:** Develop detailed plans for addressing each issue, including resource allocation and timelines.
 - **Execution:** Implement the changes with minimal disruption to users.
 - **Testing:** Test changes thoroughly in a staging environment before deployment.
 - **Documentation:** Document all changes made, including justifications and any new procedures introduced.
- 

4.6 TRAINING



Training is essential for both the maintainers and users of Dcard to ensure they are well-prepared to support and use the platform effectively. This section identifies and addresses the training needs specific to Dcard.



4.7 MAINTENANCE RECORDS AND REPORTS

🔍 **ASSISTANCE REQUESTS**


Log all requests for assistance, modifications, or problem reports.

🔍 **STATUS REPORTS**

Monitor and report the status of these requests by category and priority.

🔍 **METRICS**

Collect data on maintenance activities to evaluate performance and make data-driven decisions for future maintenance efforts.



5.0 TYPES OF MAINTENANCE






5.0 TYPES OF MAINTENANCE

1. CORRECTIVE MAINTENANCE

- Purpose: Fixes errors and bugs for a reliable application.
- Importance: Resolves issues like crashes and functional errors.
- Impact: Crucial for maintaining user satisfaction and trust.

2. ADAPTIVE MAINTENANCE

- Purpose: Modifies the system for new or changing environments.
 - Importance: Adapts to updates in operating systems, browsers, and devices.
 - Impact: Ensures compatibility and functionality across platforms.
- 



5.0 TYPES OF MAINTENANCE

3. PERFECTIVE MAINTENANCE

- Purpose: Enhances performance, usability, and functionality.
- Importance: Adds new features based on user feedback.
- Impact: Follows corrective and adaptive maintenance in priority.

4. PREVENTIVE MAINTENANCE

- Purpose: Prevents potential future problems.
- Importance: Improves reliability and prevents downtime.
- Impact: Secondary to corrective and adaptive actions.



6.0 PROGRAM COMPREHENSION

- Is essential for maintenance and evolution of large-scale systems like DCard.
- helps in maintaining, debugging, and adding new features.
- Bottom-up strategy approach to understand the system.





PROGRAM COMPREHENSION


DECONSTRUCTING

- Break down the key components such as login, registration, validation, social interaction, and content sharing.
- We will focus on functions, methods, and classes
- The purpose is to understand how the application is constructed.

ANALYZING DATA STRUCTURE

- Examined data structure for storing user info, posts, comments, and interactions.
- Insights on how these structures support basic functionality and integrate.

CONTROL FLOW

- Mapped out control flow to understand module communication.
 - The key processes are login requests and user validation.
 - Helped spot bottlenecks and areas for future improvement.
- 

PROGRAM COMPREHENSION

EXAMINING LOW-LEVEL CODE DETAILS

- Reviewed logic behind critical functions such as conditionals, loops, and error handling.
- Focused on performance optimization, stability, and bug fixes.
- Identified areas for improving efficiency and reliability.

BUILDING UP TO HIGHER LEVEL COMPONENTS

- Gradually built understanding of UI, server side processing, and database interactions.
- Formed a complete picture of the system architecture.
- Ensured changes do not disrupt existing functionality.

DOCUMENTING AND ANALYZING PATTERNS & ABSTRACTIONS

- Documented each components role and its contribution to overall functionality.
- Recorded design patterns for future maintenance and improvements.
- Ensures consistency across the system





PROGRAM COMPREHENSION

PROGRAM COMPREHENSION FOR MAINTENANCE

- Connected program understanding with maintenance tasks.
- Enabled informed decisions for improving the codebase.
- Helps in performance optimization, UI improvements, and future adaptability.

CONCLUSION

- The Bottom-up approach gave a clear perception of components, data structures, control flow, and code.
- Thorough documentation is a foundation for effective future maintenance and development.
- The goal of this is to keep DCard fast, efficient, and easily maintainable for future enhancements.





7.0 TESTING IN SOFTWARE MAINTENANCE





7.1 IF TESTING IS NOT DONE



🔍 **UNRESOLVED BUGS**

These issues may not be identified and effectively resolved, which can cause a drastic negative impact on users experience on the DCard application.

🔍 **PERFORMANCE WILL BE REDUCED**

Performance might degrade under peak load, causing slow response time and crashes.

🔍 **SOME FEATURES ARE AFFECTED**

Meet unresponsive buttons, post failing, validation issues



7.1 IF TESTING IS NOT DONE



🔍 SECURITY VULNERABILITIES

System could remain vulnerable to attacks, leaving the users data vulnerable and exposed

🔍 RELIABILITY WILL BE REDUCED

Result in unreliable and unpredictable software behavior and downtime, which will affect the overall reliability of the software.



7.2 IMPORTANCE OF TESTING



🔍 **FUNCTIONAL TESTING**

Ensures that the new and existing features of DCard application performs correctly after maintenance.

🔍 **REGRESSION TESTING**

Ensures that the new fixes do not break the existing functionalities of the application.

🔍 **PERFORMANCE TESTING**

Assesses the systems performance under load



8.0 SOFTWARE MAINTENANCE TOOLS



WHY?



- **Real-Time Performance Monitoring:**
 - Continuously tracks application metrics (response times, resource utilization, etc.)
 - Provides deep insights into runtime behavior
- **Advanced Diagnostic Capabilities:**
 - Identifies performance bottlenecks and anomalies
 - AI-driven root cause analysis for quick issue resolution
- **Enhanced User Experience:**
 - Ensures smooth and efficient application performance
 - Facilitates proactive management to avoid performance degradation



DYNATRACE



CRITERIA SELECTING DYNATRACE



🔍 **TECHNOLOGY COMPATIBILITY**

- Supports the application's technology stack (microservices, cloud environments, etc.)

🔍 **SCALABILITY**

- Handles large volumes of data and transactions efficiently

🔍 **USABILITY**

- Offers intuitive dashboards and actionable insights
- Easy for the team to interpret and use effectively



CRITERIA SELECTING DYNATRACE



🔍 **INTEGRATION CAPABILITIES**

- Seamlessly integrates with existing monitoring and management systems

🔍 **CUSTOMER SUPPORT**

- Provides reliable support and comprehensive documentation

🔍 **COST CONSIDERATIONS**

- Aligns with the organization's budget (licensing fees and total cost of ownership)

IN CONCLUSION

In conclusion, maintaining the DCard application requires addressing performance, UI, and validation issues through regular updates and optimizations. By implementing a strategic maintenance plan and utilizing tools like Dynatrace, the platform can remain stable, efficient, and user-friendly, ensuring long-term success in a competitive social networking space.

THANK YOU

Presentation by Group 6

