2025/2026 SENIOR DESIGN DESIGN DOCUMENT

Ames After Dark

Group Number: sdmay26-42
Client: Chase Anderson
Advisor: Professor Maruf
Team Email: sdmay26-42
Team Website:
https://sdmay26-42.sd.ece.iastate.edu/

Chase Anderson: Quality Control Lead (Full Stack)
Jaya Davis: Team Communications Lead (Front End)
Nathan Couture: DevOps Lead (Back End)
Nathan Krieger: Client Liaison (Full Stack)
Analyn Seeman: Project Management (SCRUM Master) (Front End)
Maggie Sullivan: Documentation Lead (Full Stack)
Geni Williams: Meeting Scribe (Full Stack)

EXECUTIVE SUMMARY

Ames After Dark is a mobile application designed to transform the nightlife experience in Ames, Iowa, by creating a centralized, real-time hub for events, bar specials, social connection, and safety. Today, students and residents rely on scattered sources like social media posts, static websites, or word-of-mouth to decide where to go out, which often results in inefficiency, missed opportunities, and safety concerns when groups get separated. This project addresses those problems by integrating event discovery, mapping, photo sharing, and location tracking into a single cross-platform mobile app.

Users will be able to quickly browse a feed of nightly events and specials, check live highlights such as the busiest bar or cheapest drinks, and apply filters to narrow their choices. An interactive map will display all Ames bars with their current open/closed status, highlight user favorites, and allow friends to share locations or plan bar crawls together. Beyond navigation, the app will provide weekly photo galleries from professional photographers, allowing students to relive and share memories, and a searchable bar directory with menus, deals, hours, and booking capabilities for events like graduations or private parties. Safety is a core consideration, with features like mutual friend location sharing, one-tap check-ins, and notification settings designed to promote responsible nightlife.

Bar owners will also benefit through a dedicated admin portal where they can upload deals, events, photos, and manage profiles, supported by role-based permissions and potential analytics in the future. The system will be built with modern, scalable technologies-React Native for the frontend, Node.js and PostgreSQL for the backend, Firebase for secure authentication, and Google Maps for geolocation-hosted on cloud free-tier services during development. While the scope of the senior design project is a working prototype rather than a production release, the deliverable will demonstrate all core features, validate technical feasibility, and provide a compelling proof-of-concept. Ultimately, Ames After Dark balances user needs for convenience, safety, and connection with bar owners' needs for visibility and engagement, offering a solution with real-world impact and strong portfolio value for the development team.

LEARNING SUMMARY

Development Standards & Practices Used

- Followed software engineering best practices including modular design, version control (GitLab), agile development, and iterative testing.
- Applied UI/UX standards for mobile applications: accessibility (contrast, text size, alt-text), and responsive layouts.
- Used security practices such as secure authentication, data validation, and role-based access for bar owners.
- Backend hosted on firebase/supabase with potential migration to AWS, real-time syncing, and API-first architecture.

Summary of Requirements

- **Must-Haves (MVP):** Navigation bar (5 tabs: Home, Map, Gallery, Bars, Account), real-time event/specials feed, interactive bar map with friend location sharing (restrictions implemented based on time and location), weekly photo galleries, bar directory (menus, deals, hours), user accounts, admin portal for bars, secure authentication, and cloud-hosted backend.
- **Should-Haves:** Push notifications, group crawl tracking, verified photographer uploads, advanced booking, notification preferences, admin analytics.
- **Could-Haves:** Trending events (based on location popularity in real time), loyalty rewards/gamification, real-time wait times or cover charges, safe-route home links.
- **Won't-Haves:** In-app ticket sales, in-app payments, public map visibility of all users, unlimited user-generated uploads, direct messaging between users.

Applicable Courses from Iowa State University Curriculum

- **COM S 309 - Software Development Practices:** agile methods, GitHub version control, full-stack teamwork
- **SE 319 - Construction of User Interfaces:** mobile-first UI/UX design, usability, accessibility

- **COM S 363 - Introduction to Databases:** relational schema design, SQL queries, data modeling
- **CPR E 308 - Operating Systems/Networking:** backend client-server interactions, cloud hosting considerations
- **SE 329 - Software Project Management:** project scoping, requirement prioritization, deliverables
- **SE 417 - Software Testing:** unit testing, integration testing, validation planning
- **CYB E 230 / CYB E 231:** cyber security fundamentals, attack prevention tools, and network security methods

New Skills/Knowledge Acquired that was not taught in Courses

- **Mobile cross-platform development** with React Native.
- **Real-world cloud deployment practices** using Firebase/Supabase and evaluating scalability on AWS.
- **Integration of third-party APIs** (Maps, Firebase Authentication) and handling geolocation.
- **Admin portal design** for non-technical users, including CRUD tools and role-based permissions.
- **Practical UI prototyping** in Figma and translation into functional mobile code.
- **Domain-specific knowledge** of bar/nightlife operations in Ames, including specials, events, and booking workflows.

TABLE OF CONTENTS

LIST OF FIGURES

No figures are included in the document yet.

1. INTRODUCTION

1.0 Problem Statement

Nightlife in Ames, Iowa, plays a big role in student and community life, but finding accurate, up-to-date information about what is happening is harder than it should be. Students often learn about bar specials, live music, or events through scattered Instagram posts, flyers, or word-of-mouth. This makes it easy to miss out on opportunities, wastes time trying to decide where to go, and sometimes creates safety issues when groups get separated late at night. There is no single platform that brings all of this information together while also supporting the social and safety needs of the people who participate in Ames nightlife. Ames After Dark seeks to address this gap by offering a one-stop mobile app where users can discover real-time events and deals, connect with friends on an interactive map, and stay safe while enjoying the nightlife scene. By creating a more connected and informed experience, the app not only makes going out easier and more fun but also supports safety and community in Ames.

1.1 Intended Users

The primary users of Ames After Dark are college students at Iowa State University and nearby schools. These students are highly social, often go out in groups, and want quick, reliable ways to find the best events or deals on any given night. Their needs include convenience (finding specials fast), connection (staying in touch with friends on bar crawls), and safety (knowing where their group is and having quick access to check-ins).

A second key group is Ames residents who enjoy nightlife but are not students. These users may be young professionals, alumni, or locals who want to explore the bar scene without relying on word-of-mouth. Their needs are slightly different: they may prioritize reliable information about events, bar hours, or booking spaces for special occasions like birthdays or graduations.

Finally, bar owners and event organizers are also important users. They benefit by having a direct way to share deals, events, and photos with a targeted audience. Their needs include easy-to-use tools to update specials, manage bar profiles, and attract more customers. By giving bars a centralized platform, Ames After Dark not

only supports their business goals but also ensures that the information students and residents see is timely and accurate.

Together, these groups highlight the dual value of the app: making nightlife in Ames more fun and safer for users, while also strengthening the connection between bars and their community.

2. REQUIREMENTS, CONSTRAINTS, AND STANDARDS

2.1 Requirements & Constraints

2.1.1 Functional Requirements

Functional requirements define the essential behaviors and capabilities of the system. These requirements are directly tied to user needs such as event discovery, social connection, and nightlife safety.

- Provide a real-time feed of nightly events and drink specials.
- Include an interactive map of Ames bars with open/closed status and friend locations (mutual, opt-in).
- Support a searchable bar directory with menus, specials, and hours, with the ability to favorite/unfavorite bars.
- Provide weekly photo galleries uploaded by verified photographers, organized by bar and date, with download/share functionality.
- Support secure user registration, login, profile management, and friend connections.
- Provide an admin portal where bar owners can upload deals, events, and photos, with role-based permissions.
- Maintain cloud-hosted backend and relational database with real-time syncing.

2.1.2 User Experience Requirements

User experience requirements describe qualities that affect how users interact with the system, focusing on convenience, usability, and safety.

- Update bar open/closed status at least once per hour.
- Provide filtering options for events, specials, and bar availability.

- Restrict location sharing to mutual friends and explicit opt-in.
- Allow users to browse and filter photo galleries by bar and date.
- Enable one-tap interactions for common actions (e.g., checking in, toggling location visibility).

2.1.3 Physical / Resource Requirements

These requirements specify the platforms, infrastructure, and resources needed to implement and operate the system.

- Must run on both iOS and Android platforms.
- Backend hosted on cloud services using free-tier plans during development.
- Use a relational database (e.g., PostgreSQL) to store bar data, user accounts, and event information.
- Integrate with third-party APIs for maps and authentication.

2.1.4 Aesthetic / UI Requirements

Aesthetic and UI requirements ensure the interface is visually clear, consistent, and usable in fast-paced nightlife environments.

- Use a bottom navigation bar with five core sections: Home, Map, Gallery, Bars, and Account.
- Fonts and color contrasts must follow accessibility guidelines (e.g., WCAG minimum contrast ratios).
- Visual layout should prioritize quick, one-handed interactions and readability in low-light environments.
- Use consistent iconography and theming across all screens.

2.1.5 Constraints

Constraints represent boundaries and limitations that affect design decisions.

- The system will not include in-app ticket sales or payments.
- User location visibility restricted to mutual friends only; no public tracking.
- Initial events and deal data may be manually curated rather than fully automated.
- Development is limited to the senior design two-semester timeframe; production-level scaling is not feasible.
- Must rely on free-tier cloud hosting during senior design development.

2.2 Engineering Standards

Engineering standards are important in ensuring quality, interoperability, and usability in software projects. For Ames After Dark, applying the right standards helps the team avoid vague requirements, design accessible interfaces, and ensure reliable system behavior. The following standards were selected as most relevant to our project:

2.2.1 ISO/IEC/IEEE 29148:2018 (Systems and software engineering – Life cycle processes – Requirements engineering)

- **Description**: This standard provides a framework for effective requirements engineering across the system life cycle. It outlines how to write high quality requirements, ensure they are consistent and verifiable, and manage them as projects evolve. It also defines attributes of good requirements, such as being clear, feasible, and testable.
- **Relevance**: For Ames After Dark, ISO/IEC/IEEE 29148:2018 helps the team create precise and measurable requirements. Instead of vague goals like "the app should be fast," this standard guides us to specify requirements such as "bar status must update within one minute." Using this approach improves clarity for developers, ensures testability, and reduces the risk of misunderstandings during design.

2.2.2 ISO/IEC 27001:2022 (Information security, cybersecurity and privacy protection – Information security management systems – Requirements)

- **Description**: This standard defines requirements for establishing, implementing, and continually improving an Information Security Management System (ISMS). It aims to protect confidentiality, integrity, and availability of data and includes controls such as access control, encryption, auditing, and secure development.
- **Relevance**: Because Ames After Dark handles user accounts, location sharing, and possibly sensitive user data, ISO/IEC 27001:2022 helps define how we protect that data. For example, we'd use access controls, encryption of sensitive data, and more to protect all sensitive data. Applying this standard helps us better defend against breaches and build trust with users.

2.2.3 ISO/IEC 25010:2023 (Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Product quality model)

- **Description**: This standard defines a comprehensive model for evaluating the quality of Information and Communication Technology (ICT) products and software. It identifies nine characteristics including functional suitability, performance efficiency, compatibility, interaction capability, reliability, security, maintainability, flexibility, and safety. The standard provides a reference model for specifying, measuring, and evaluating product quality throughout its lifecycle, addressing how well a product meets each quality attribute.
- **Relevance**: For Ames After Dark, ISO/IEC 25010:2023 provides a structured framework for defining and evaluating software quality across multiple dimensions. By referencing this model, the team can set clear targets for performance efficiency (real-time event updates), reliability (consistent uptime), and interaction capability (intuitive mobile navigation). The inclusion of safety and security aspects also supports responsible handling of user location and data. Applying this standard ensures that the app not only functions correctly but also meet high expectations for quality, usability, and long-term maintainability.

3. PROJECT PLAN

3.1 Project Management/Tracking Procedures

Our team is adopting an Agile development approach, emphasizing iterative progress, flexibility, and regular communication. Because Ames After Dark is a complex application involving both frontend and backend development, Agile allows the team to deliver functionality in manageable increments, respond to feedback, and adapt design decisions as new requirements emerge. Each sprint will focus on completing a subset of prioritized features and refining them through testing and review.

To track progress and coordinate effectively, the team uses the following tools and methods:

- Gitlab – used for version control, issue tracking, and merge request management.
- Microsoft Teams – used for file sharing, meeting information, and longer technical messages and questions.
- Group SMS Chat – used for fast updates, reminders, and coordinating meeting times or small decisions.
- Face-to-Face Meetings – used during scheduled team work sessions to discuss major updates, plan tasks, and resolve technical or design issues collaboratively.
- Email – used to communicate formally with the advisor, particularly for scheduling and milestone updates.

The combination of these tools supports transparency, accountability, and continuous progress tracking throughout both semesters.

3.2 Task Decomposition

The Ames After Dark system is divided into several high-level components, each with associated tasks and subtasks. This decomposition supports parallel development and clearer ownership of deliverables.

1. Project Setup and Planning
    a. Define requirements and constraints
    b. Select technology stack (frontend, backend, hosting, APIs)
    c. Create development repositories and documentation structure
2. Frontend Development
    a. Design and implement navigation bar and screen layouts
    b. Develop core screens (Home, Map, Gallery, Bars, Friends)
    c. Implement photo gallery view with filters
    d. Integrate authentication and location permissions
3. Backend Development
    a. Design and implement database schema
    b. Develop APIs for events, bar data, and authentication
    c. Configure Firebase for secure login and storage
    d. Manage real-time syncing
4. Integration and Testing
    a. Connect frontend with backend APIs
    b. Conduct unit and integration testing
    c. Address performance bottlenecks and security issues
5. Deployment and Documentation

a. Deploy prototype to beta testers/test devices
b. Prepare final documentation and presentation

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

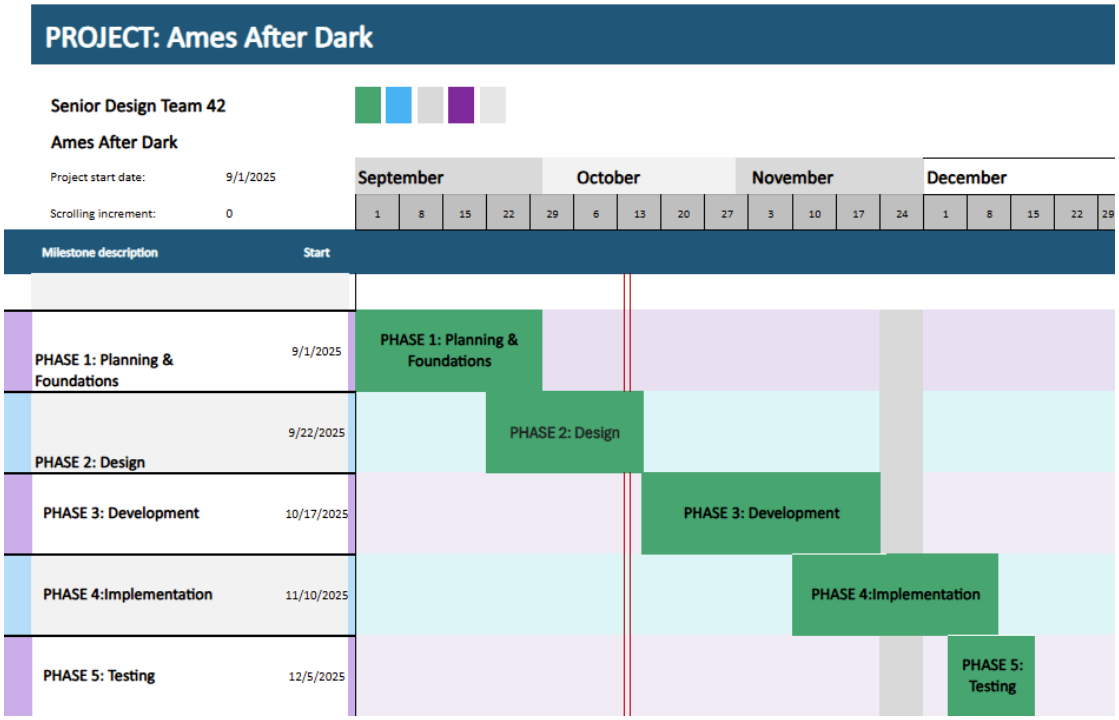
Key milestones are established to measure progress and ensure timely completion. Each milestone corresponds to functional goals or sprint deliverables.

Milestone	Target Completion	Metric / Evaluation Criteria
Screen Finalization	September 2025	Team consensus on finalized UI layout and navigation flow in Figma; approval from advisor and client.
ER Diagram & Database Design	October 2025	Logical and physical ERD completed, reviewed, and approved; database schema accurately represents all major entities and relationships.
Design Documentation Completion	October 2025	Design document sections drafted, internally reviewed, and integrated with minor revisions.
Technology Stack Confirmation	October 2025	Full team agreement on frameworks, APIs, and hosting platforms; compatibility validated on all developer machines.
Project Requirements Lock	October 2025	Functional and non-functional requirements finalized, reviewed by team, and approved by client.
Frontend Structure Implemented	November 2025	Navigation and five core screens (Tonight, Map, Gallery, Bars, Account) operational in Expo simulator.
Backend API and Database Operational	November 2025	CRUD operations for users, bars, events, and photos verified through Postman; live connection to Supabase confirmed.
Integration Between Frontend and Backend	December 2025	Data retrieved and displayed in app without major errors; successful end-to-end tests for login, bar listings, and photo gallery.
User Testing Round 1 – Internal Team	February 2026	Internal test plan executed; >90 % of major functionality verified by development team; bugs logged and prioritized.

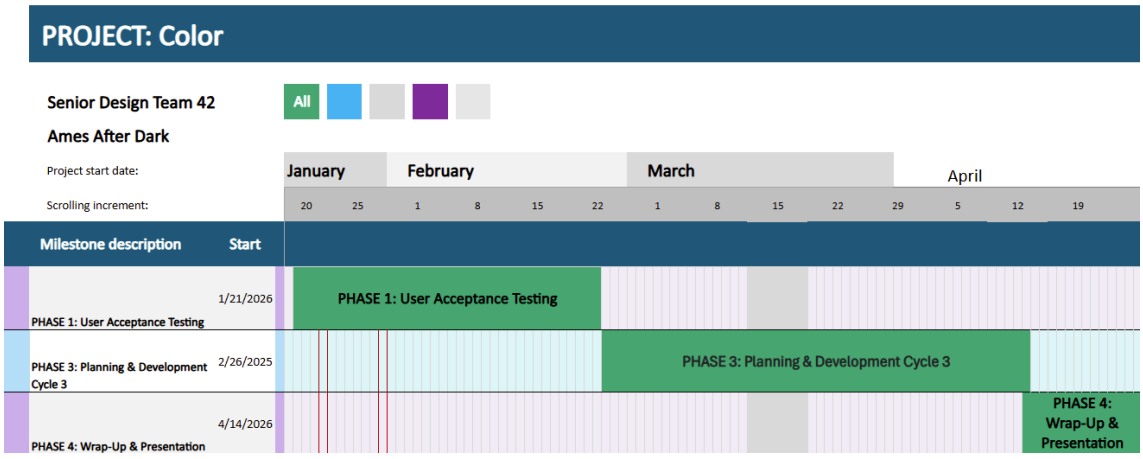
User Testing Round 2 – Controlled Beta Group	March 2026	10–15 student testers complete structured usability session; feedback aggregated and ≥80 % satisfaction rating achieved.
Advanced Feature Integration	March 2026	Friend map updates, real-time specials, and admin uploads implemented; regression tests pass with <5 % critical failures.
User Testing Round 3 – Live Bar Field Testing	April 2026	Successful pilot with 2–3 local bars; patrons can view live data and photos; app stability ≥95 % uptime during live use.
Final Fixes and Polish	Mid-April 2026	All critical bugs resolved; UI consistent with design standards; verified through advisor review.
Final Documentation and Presentation	Late April 2026	Technical report, user guide, and poster complete; live demonstration to faculty with all MVP features fully functional.

3.4 Project Timeline/Schedule

Semester 1



Semester 2



Detailed Schedules

This is a summary of the project timeline altered for readability. View the full Gantt chart here:

- Full Gantt Charts

3.5 Risks and Risk Management/Mitigation

Risk	Probability	Mitigation Strategy
Integration issues between frontend and backend	0.6	Conduct early API testing; maintain consistent data structures across team
API rate limits or authentication errors	0.4	Implement caching and fallback data for critical features
Schedule delays due to exams/holidays	0.4	Plan lighter sprints around academic breaks
Security vulnerabilities (data leaks or poor access control)	0.3	Follow ISO/IEC 27001 principles, apply secure login and encryption
Cloud service limits (free-tier restrictions)	0.5	Optimize data storage and monitor usage; migrate if necessary

3.6 Personnel Effort Requirements

The table below estimates the total person-hours required for major tasks in the Ames After Dark project. Estimates are based on relative task complexity, expected learning curves, and each member's role specialization. These numbers reflect projected effort across both semesters of senior design.

Task	Team Members	Estimate Hours	Description

Project planning and documentation	All	60	Define project scope, requirements, constraints, and maintain documentation.
Frontend Development	Jaya, Analyn, Maggie, Chase, Nathan K, Nate C	160	Develop main app screens, implement navigation, and integrate visual components with backend APIs.
Backend Development	Geni, Nate C, Chase, Nathan K	170	Set up database schema, implement API endpoints, and establish data synchronization.
Integration and Testing	All	100	Perform unit, integration, and user testing; address bugs and verify system functionality across platforms.
Deployment and Hosting Setup	Nate C, Geni	40	Configure cloud hosting and app distribution for testing
Final Presentation and Deliverables	All	50	Prepare final demo, documentation, and presentation materials for faculty.

3.7 Other Resource Requirements

Aside from personnel effort and time, the Ames After Dark project relies on several external and technical resources necessary for successful completion.

Resource	**Type**	**Purpose/Description**
Bar Participation	External Partner	Collaboration with local Ames bars to gather accurate event, deal, and profile data for testing.
Photographer Participation	External Partner	Work with bar photographers to source sample gallery images and verify the upload and tagging process.
Cloud Hosting	Technical	Provide authentication, database storage, and real-time synchronization for user and bar data.
Google Maps API	Technical	Enable location-based services including friend sharing and bar mapping.
Development Tools	Technical	Includes Visual Studio Code, GitLab, Figma, and Teams for development, version control, design, and communication.
Moile Devices (iOS/Android)	Hardware	Used for testing app performance and interface compatibility across platforms.

University Resources	Institutional	Access to ISU's network, software licenses, and senior design facilities for collaboration and testing.

These resources ensure the team can simulate realistic use cases, meet accessibility and reliability goals, and deliver a functional prototype by the end of the project timeline.