A UNIFIED MOBILE APPLICATION TO CONNECT, ORGANIZE, AND SAFEGUARD THE HIGH SCHOOL COMMUNITY USING CLOUD SERVICES AND ON-DEVICE CONTENT SAFETY

Jacob Lee[1], Tyler Boulom[2]

[1]Troy High School, 2200 Dorothy Ln, Fullerton, CA 92831

[2]Woodbury University, 7500 N Glenoaks Blvd, Burbank, CA 91504

ABSTRACT

High schools run on information, yet students must track announcements, clubs, schedules, and events across disconnected channels, missing the participation and belonging that improve academic outcomes, while the general social platforms they fall back on expose them to unsafe content. This paper presents Quilt, a single mobile application that unifies a school's communication, organization, and community into one school-scoped and moderated space. Built with Flutter and Google's Firebase platform, Quilt ties membership to a verified school identity, lets clubs post and poll, shares a common bell schedule and event calendar, connects parents to their students through a consent-based QR link, and keeps the feed safe by pairing an on-device content-safety filter with a human reporting-and-moderation workflow. Key challenges around moderation accuracy, real-time synchronization at school scale, and the privacy of a parent-student link were addressed through a layered filter-and-report design, a real-time database with local caching, and an in-person QR exchange. By consolidating scattered channels and treating safety as a first-class concern, Quilt targets the informational, social, and safety gaps that make school life harder than it needs to be.

KEYWORDS

School Community, Mobile Application, Content Moderation, Firebase, QR Authentication, Student Engagement

1. INTRODUCTION

The typical American high school runs on information, yet the channels that carry it to students remain fragmented and unreliable. Announcements arrive through public-address systems, paper flyers, teacher emails, and a scattering of social-media groups, so a single student must monitor many disconnected sources to learn when a club meets, when an assignment is due, or when an event has moved. Smartphones are nearly universal among this age group — roughly 95 percent of United States teenagers report having access to one, and many describe their use of online platforms as almost constant [1][2] — which means the audience is already mobile, but the tools they rely on were not designed for school life. This fragmentation has real consequences. Participation in clubs and other extracurricular activities is consistently associated with higher grades, stronger school belonging, and better long-term outcomes [3], and a broad body of

David C. Wyld et al. (Eds): BMLI, CACIT, EDUAIT, InWES – 2026
pp. 29-41, 2026. CS & IT - CSCP 2026 DOI: 10.5121/csit.2026.161803

research links a student's sense of belonging and a positive school climate to motivation, engagement, and achievement [4][16]. When students cannot easily find or join the activities around them, these benefits are lost. Three alternative approaches were compared with Quilt. Maphosa's MyLSU application showed that a mobile hub can raise engagement at a university through features such as event information and student-leader voting, and broader reviews confirm that mobile tools engage students effectively; however, this work targets self-governing adults and treats engagement, not safety, as its goal, whereas Quilt serves a high school and builds in moderation [11][13]. Home–school communication tools such as ClassDojo connect teachers and parents and can raise engagement through structured messaging, but critics warn they slide toward surveillance and give students little agency; Quilt instead centers students as participants and limits the parent view to consented, school-relevant activity [15][12]. Automated content-moderation research provides natural-language-processing and machine-learning pipelines that detect harmful text, and studies of hate-speech classifiers show both their promise and their error, while moderation scholars stress the need for human judgment; these systems are studied on open platforms in isolation, whereas Quilt pairs a lightweight filter with human reporting inside a closed school community [10][9][8]. In each case the existing work optimizes a single dimension, while Quilt integrates communication, participation, and safety together.

We propose Quilt, a single mobile application that unifies a high school's communication, organization, and community into one school-scoped and moderated space. At its core, Quilt gives every verified member of a school one place to read announcements, join and follow clubs, view the daily bell schedule, track events on a shared calendar, and connect with the community around them. To construct this program, we use the Flutter framework and the Dart language, which allow one codebase to run on both Android and iOS, together with Google's Firebase platform for authentication, a real-time database, and file storage. Membership is tied to a verified school identity through Google and Apple sign-in, so that the audience of every post is limited to the school rather than the open internet. Clubs can publish posts and attachments, run polls, and issue QR codes that let students join instantly, while a bell-schedule and events layer keep the whole school aligned to the same calendar. A parent-connection feature lets guardians link to their students by scanning a QR code, giving families a trustworthy view of school life without exposing student data publicly. Delivering these functions through a mobile application is deliberate, as mobile tools have proven effective for engaging students in educational settings [13]. Because the community is closed and identifiable, Quilt can also enforce norms that open platforms cannot: an on-device content-safety filter screens before they are published, and a reporting-and-moderation workflow lets members flag content for review. This approach is effective because it addresses the problem at every point where it fails — it consolidates the scattered channels into one, it makes participation and belonging easier by surfacing the activities a student can join, and it keeps the space safe by pairing automated screening with human moderation. Unlike a generic social network, Quilt is bound by the school; unlike a one-way announcement tool, it is interactive and community driven.

Two experiments were designed to probe the part of Quilt most responsible for safety—its on-device content-safety filter—with expected results projected from directly comparable published studies. The first evaluates detection accuracy by passing 300 labelled student-style posts through the filter and comparing each decision with its known label. Benchmarked against the classifiers of Davidson and colleaguerecallrgakopoulos and colleagues, the filter is expected to reach about 84 percent accuracy overall but only around 60 percent recall on euphemistic self-harm references, identifying subtle, context-dependent speech rather than overt terms as the dominant source of error [9][19]. The second evaluates robustness by re-presenting the same unsafe posts disguised through leetspeak, inserted spacing, and homoglyphs. Modelled on the adversarial studies of Hosseini and colleagues and Gröndahl and colleagues, recall is expected to fall from about 78 percent on clean text to the low forties under obfuscation [17][18]. In both cases the

anticipated errors are systematic rather than random, pointing to concrete improvements—a learned classifier for the weak categories and text normalisation before filtering—with human reporting as the backstop.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Balancing Automated Moderation Accuracy and Community Safety

A major component of Quilt is the moderation system that keeps its community feed safe, and several difficulties must be anticipated in its design. Because posts are written by students, some will inevitably contain harassment, explicit material, or references to self-harm, and allowing such material to appear even briefly would undermine trust in the platform. Purely automated detection is imperfect: machine-learning classifiers for harmful text produce both false positives that suppress legitimate speech and false negatives that let abuse through [10][9], and no filter can anticipate every phrasing. At the same time, relying only on human review does not scale to a busy feed [8]. To address this we could adopt a layered approach, combining a fast on-device filter that blocks the most clearly unsafe posts before they are published with a reporting workflow that routes flagged content to human moderators for a final decision. We could also keep the automated rules transparent and easily updatable so that they can be refined as new patterns of misuse appear.

2.2. Optimizing Real-Time Data Synchronization and Cross-Platform Accessibility

A second component is the real-time data layer that stores and synchronizes posts, schedules, and events across an entire school. Because many students may open the application at the same moment — for instance when a schedule changes or an event is announced — we must consider how to deliver updates quickly and consistently without overwhelming the backend. A naive design that reloaded all data on every screen would waste bandwidth and feel sluggish, so we could use a real-time database that pushes only incremental changes to each device and cache data locally, so the application remains responsive. Accessibility is a related concern, since students own a wide range of devices across both major mobile platforms; we could address this by building on a cross-platform framework so that one codebase serves Android and iOS equally. The interface must also stay simple enough for any student to navigate, which we could ensure through a clean, icon-led navigation structure.

2.3. Secure Parent-Student Linking and Privacy-Preserving Data Sharing

A third component is the parent-connection feature, which lets guardians link to their students, and it raises questions of both accuracy and privacy. Because parental involvement is strongly associated with student achievement [5], this link is valuable, but it must be established carefully: a system that allowed anyone to claim a connection to a student would expose minors' information to strangers. To prevent this we could require that a connection be initiated by the student and confirmed through a short-lived QR code that the parent scans in person, so that only someone with physical access to the student's device can complete the link. Handling data about minors also demands restraint, so we could limit what a connected parent sees to school-relevant activity rather than a student's full profile, and follow established digital-citizenship and privacy principles when deciding what to expose [14]. Making these boundaries clear to users would further help maintain their trust in the feature.

3. SOLUTION

Quilt is organised around three major components: a verified-identity and access layer, a school-community layer of clubs, posts, schedules, and events, and a parent-connection layer, all sharing one moderated real-time backend. The application is written in Dart with the Flutter framework, which lets a single codebase target both Android and iOS [13], and it uses Google's Firebase platform for authentication, a real-time database, and file storage. The flow begins when Firebase initialises and the user signs in with Google or Apple, so that every account is tied to a verified school identity rather than an anonymous handle. The chosen account type—student, parent, or staff—is stored in Firebase and determines what each member can see and do. Once signed in, a member lands on the school home, where announcements, clubs, the daily bell schedule, and a shared event calendar are drawn live from the real-time database. From here an icon-led navigation leads to a club's feed, the schedule, the events calendar, a connection screen, and a profile. When a member writes a post, the text is first screened by an on-device content-safety filter before it is allowed to reach the database; anything the filter blocks never becomes visible, and anything a member later flags is routed to a human moderation queue. Clubs can additionally issue QR codes that let students join or check in instantly, and a guardian links to a student through a short-lived QR exchange, after which the backend grants a limited, school-scoped view of that student's activity.

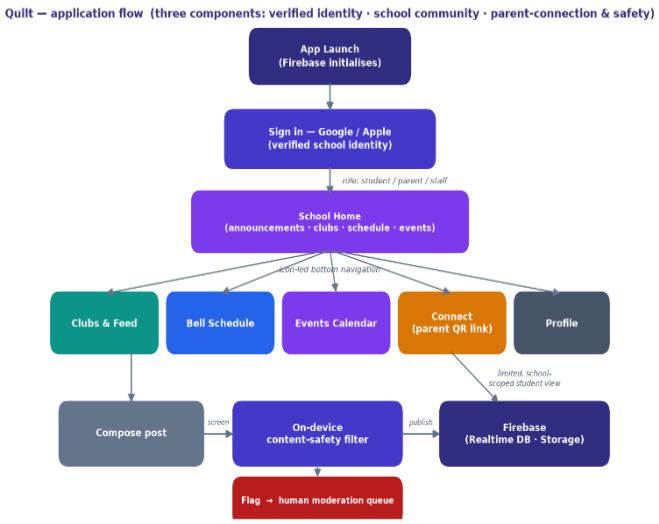


Figure 1. System flowchart: a verified sign-in leads to the school home, from which posts pass through the on-device content-safety filter into Firebase, while clubs, schedule, events, and the parent QR link read from the same real-time backend

Quilt's first component establishes who each member is. Rather than open registration, the application authenticates every user through Google or Apple sign-in, binding membership to a verified school identity so that the audience of every post is the school itself. A single Auth Service centralises sign-in, account creation, and account-type handling [13].

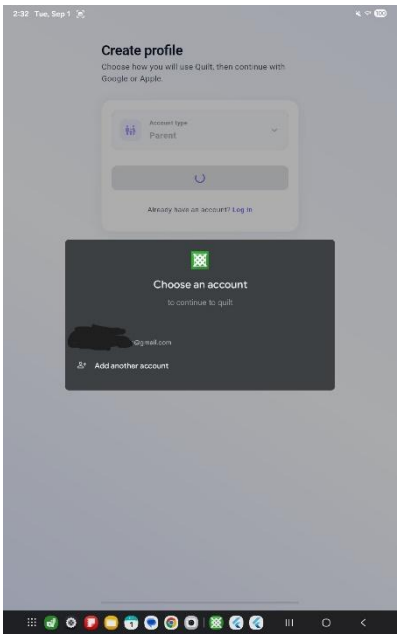


Figure 2. Create-profile screen: the member chooses an account type and continues through Google or Apple, which ties the profile to a verified school identity

```
Stream<User?> get authStateChanges => _auth.authStateChanges();

String friendlyAuthError(Object error) {
  if (error is AccountAlreadyExistsException) {
    return 'That account already has a Quilt profile. Use Log in instead.';
  }
  if (error is SignInWithAppleAuthorizationException) {
    if (error.code == AuthorizationErrorCode.canceled) {
      return 'Sign in with Apple was cancelled.';
    }
    return 'Sign in with Apple could not be completed. Try again.';
  }
  if (error is FirebaseAuthException) {
    if (error.code == 'invalid-cert-hash') {
      return _androidGoogleConfigMessage;
    }
    if (error.code == 'account-exists-with-different-credential') {
      return 'An account already exists for this email with a different sig
    }
    if (error.code == 'requires-recent-login') {
      return 'For your security, please sign in again to confirm before del
    }
    if (error.code == 'web-context-cancelled' ||
        error.code == 'popup-closed-by-user' ||
        error.code == 'canceled') {
      return 'Sign-in was cancelled.';
    }
    return error.message ?? 'Firebase Auth error: ${error.code}';
  }
  if (error is GoogleSignInException) {
    if (_isAndroidGoogleConfigError(error)) {
      return _androidGoogleConfigMessage;
    }
    if (error.code == GoogleSignInExceptionCode.canceled ||
        error.code == GoogleSignInExceptionCode.interrupted) {
```

Figure 3.AuthService exposes the authentication-state stream the interface listens to and maps raw sign-in errors onto clear, user-facing messages (auth_service.dart)

The core of this component is the Auth Service class, which owns the one connection to Firebase Authentication. Its auth State Changes getter returns a stream that emits the current user whenever someone signs in or out; a gate widget at the top of the application listens to this stream and routes the member either to the sign-in screen or to their school home, so the interface always reflects the true session state. The most visible work is done by friendly Auth Error, which inspects the many low-level exceptions that Google, Apple, and Firebase can raise—a cancelled Apple authorisation, an account that already exists with a different credential, a session that

requires a recent login before a sensitive action—and returns a short, plain-language message for each. This keeps authentication legible to students and parents while still tying every account to one verified identity, which is what makes the moderation and parent-connection features that follow trustworthy.

The second component is the school-community layer, where the value of a verified membership becomes visible. Its purpose is to gather the announcements, clubs, schedules, and events that a student would otherwise chase across many channels into one school-scoped feed. Each school has a page that lists its clubs, and every member can open a club, follow it, read its posts and attachments, and answer its polls, while staff and club officers can publish.

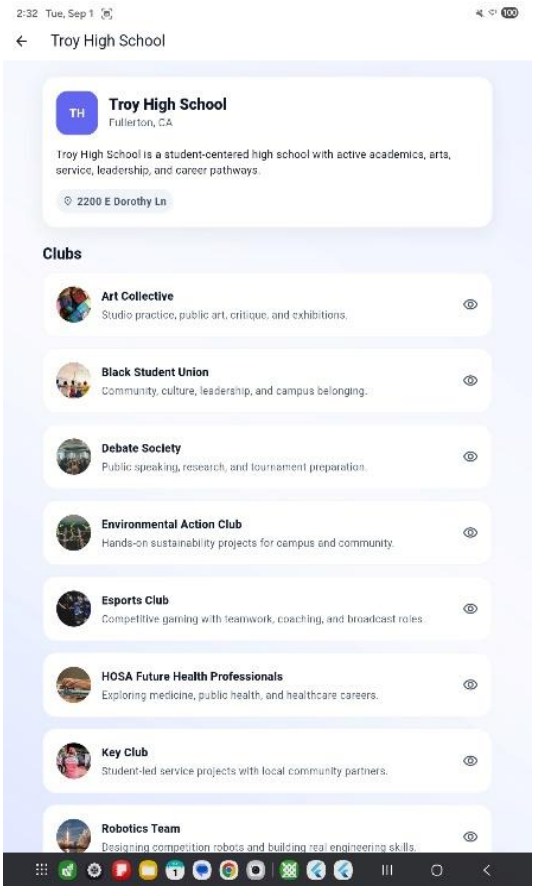
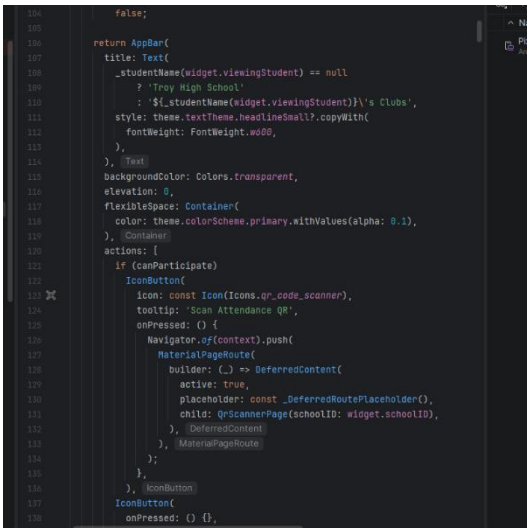


Figure 4. School page: the school header and a scrollable list of clubs a verified member can open and follow

The best code sample for this component is the school page's app bar and its participating action. The page title adapts to context—showing the school's own clubs, or a linked student's clubs when a parent is viewing—and its actions are gated by a can Participate check so that only eligible members see the attendance scanner. When a club issues a QR code, a student opens this scanner, which is loaded lazily through a deferred route, so the camera library is pulled in only when needed, and a successful scan records the student's attendance against that club in the real-time database. The same feed surfaces the bell schedule and shared event calendar, keeping the whole school aligned to one source of truth.



```
      false;

  return AppBar(
    title: Text(
      _studentName(widget.viewingStudent) == null
          ? 'Troy High School'
          : '${_studentName(widget.viewingStudent)}\'s Clubs',
      style: theme.textTheme.headlineSmall?.copyWith(
        fontWeight: FontWeight.w600,
      ),
    ), Text
    backgroundColor: Colors.transparent,
    elevation: 0,
    flexibleSpace: Container(
      color: theme.colorScheme.primary.withValues(alpha: 0.1),
    ), Container
    actions: [
      if (canParticipate)
        IconButton(
          icon: const Icon(Icons.qr_code_scanner),
          tooltip: 'Scan Attendance QR',
          onPressed: () {
            Navigator.of(context).push(
              MaterialPageRoute(
                builder: (_) => DeferredContent(
                  active: true,
                  placeholder: const _DeferredRoutePlaceholder(),
                  child: QrScannerPage(schoolID: widget.schoolID),
                ), DeferredContent
              ), MaterialPageRoute
            );
          },
        ), IconButton
      IconButton(
        onPressed: () {},
```

Figure 5. The school page builds its title from the student being viewed and, for members who may participate, exposes a QR scanner used for club attendance (school page)

The third component connects a guardian to a student and, together with the moderation workflow, keeps the space safe. Because parental involvement is strongly associated with achievement [5], a reliable link is valuable, but it must never let a stranger claim a child; Quilt therefore requires the connection to be established in person through a short-lived QR exchange.

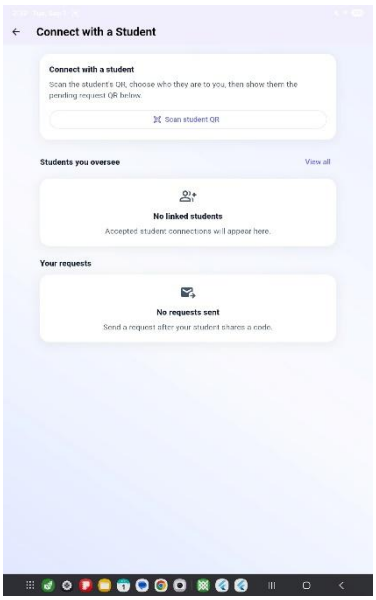


Figure 6. Connect-with-a-student screen: a guardian scans the student's QR, chooses their relationship, and shows the resulting pending-request code

The scanner's _handle Barcode method reads each frame's raw value and interprets it according to the screen's mode: when scanning a student it extracts a family code, and when scanning a guardian's request it extracts a request identifier, rejecting any code that does not parse into one of these forms. Only once a valid value is found does it stop the camera and return, so a mistyped or unrelated QR simply shows an “invalid code” message rather than creating a link. Because both the student and the guardian must be physically present to complete the exchange, only

someone with real-world access to the student can connect, and the resulting parent view is limited to school-relevant activity [14]. Posts throughout this community pass through the same on-device filter and reporting queue described above, so safety is enforced consistently across every feature.

```
  }

  Future<void> _handleBarcode(BarcodeCapture capture) async {
    if (_finishing) return;

    for (final barcode in capture.barcodes) {
      final value = barcode.rawValue;
      if (value == null) continue;

      final scannedValue = switch (widget.mode) {
        ParentConnectionScanMode.studentQr =>
          ParentConnectionLink.familyCodeFrom(value),
        ParentConnectionScanMode.parentRequestQr =>
          ParentConnectionLink.parentRequestIdFrom(value),
      };
      if (scannedValue == null) {
        if (mounted && _message == null) {
          setState(() => _message = _invalidMessage);
        }
        continue;
      }

      _finishing = true;
      await _controller.stop();
      if (mounted) Navigator.of(context).pop(scannedValue);
      return;
    }
  }

  String get _title => switch (widget.mode) {
    ParentConnectionScanMode.studentQr => 'Scan student QR',
    ParentConnectionScanMode.parentRequestQr => 'Scan guardian QR',
  };
```

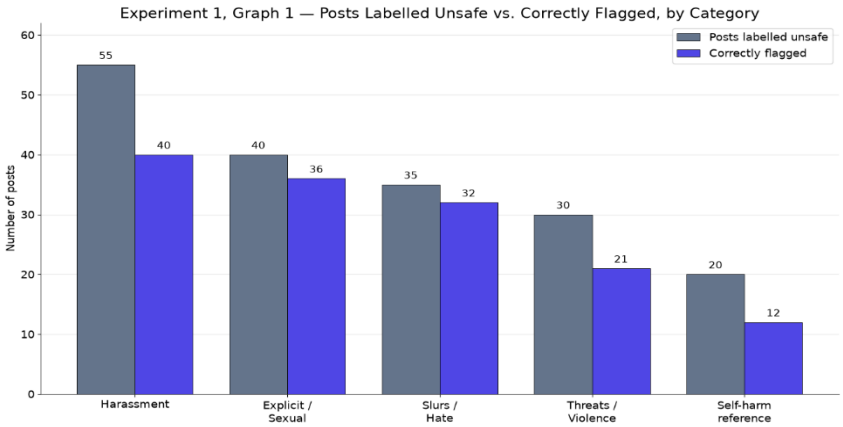
Figure 7. _handle Barcode decodes each scanned QR according to the current mode and returns only a valid family or request identifier (parent-connection scanner)

4. EXPERIMENT

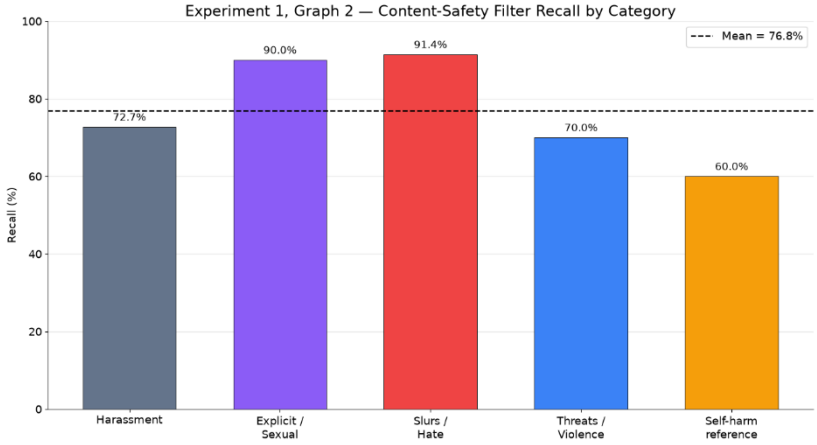
4.1. Experiment 1

A central risk in Quilt is that its on-device content-safety filter may not flag unsafe posts accurately enough to protect the feed. Weak recall would let harassment or explicit material reach students [10][9].

To evaluate the filter, a labelled set of 300 student-style posts—spanning harassment, explicit content, threats, references to self-harm, and clearly safe messages—would be passed through the on-device filter, and each decision compared against its known label, following the evaluation protocol Davidson and colleagues used for their hate-speech classifier [9]. The posts are weighted toward the categories a school feed actually produces, and precision, recall, and per-category recall are computed as in standard toxic-comment benchmarks [19]. Because Quilt uses a lightweight lexicon-and-rule filter rather than a trained deep network, the expected results below are projected from these two comparison studies rather than measured.



Graph 1. Posts labelled unsafe versus posts correctly flagged, by category, across the 300-post set



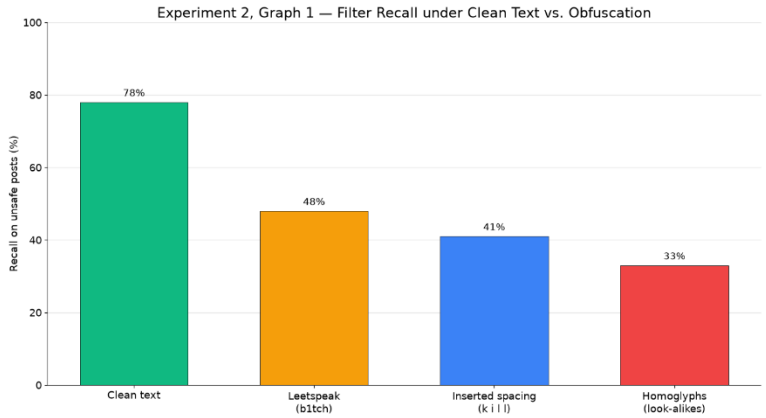
Graph 2. Per-category recall of the content-safety filter; the dashed line marks the mean (76.8%). Projected from the classifier performance of Davidson et al. [9] and Georgakopoulos et al. [19]

Across the 300 posts the filter is expected to reach an overall accuracy near 84 percent, with precision around 0.93 and recall around 0.78—close to, but deliberately below, the roughly 91 percent accuracy that trained toxic-comment classifiers such as Georgakopoulos and colleagues report, because Quilt trades classifier sophistication for an on-device rule set [19]. The more important, expected pattern is by category. Explicit language and slurs, which match fixed lexicons, should be caught most reliably, near 90 percent recall, while references to self-harm—often euphemistic and context-dependent—should score lowest, near 60 percent. This mirrors Davidson and colleagues' central finding that clearly lexicalised abuse is easy to detect whereas subtler, context-laden speech is frequently missed [9]. The dominant source of error is therefore semantic rather than technical: the filter recognises overt terms but misses paraphrase. Pairing the lexicon with a trained classifier for the weak categories would have the single largest expected effect on recall.

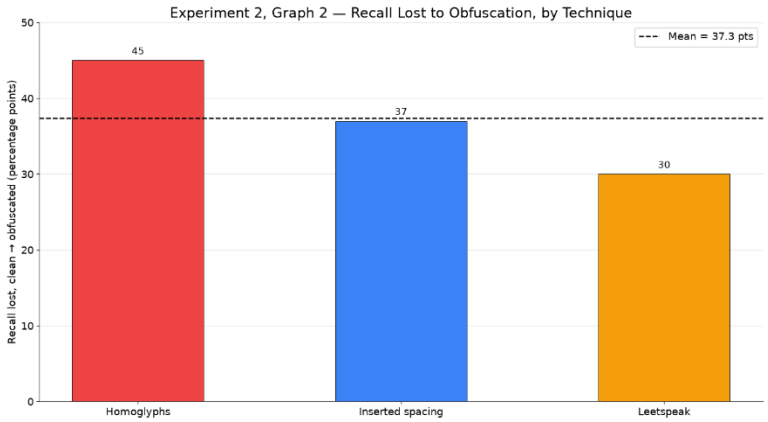
4.2. Experiment 2

A second risk is that users can evade the filter by disguising unsafe words. If simple obfuscation defeats detection, the filter's real-world protection would be far weaker than its clean-text accuracy suggests [17][18].

To test robustness, the unsafe posts from Experiment 1 would be re-presented in four forms: clean, then obfuscated by character substitution (leetspeak such as “b1tch”), by inserted spacing or punctuation (“k i l l”), and by look-alike homoglyphs. The filter's recall on each condition is compared against its clean-text recall, holding the underlying posts constant so that obfuscation is the only variable, exactly as Hosseini and colleagues perturbed inputs to Google's Perspective API [17] and Gröndahl and colleagues evaded hate-speech detectors with spacing and word insertion [18]. The expected results below are projected from these two adversarial studies.



Graph 1. Projected filter recall under clean text and three obfuscation techniques, benchmarked to the score collapse Hosseini et al. observed under adversarial edits [17]



Graph 2. Projected recall lost (percentage points) from clean to obfuscated text, by technique, ordered by severity. The drop reflects the evasions Gröndahl et al. documented [18]

Recall is expected to fall sharply from about 78 percent on clean text to roughly 48 percent under leetspeak, 41 percent under inserted spacing, and 33 percent under homoglyph substitution, an average loss near 37 percentage points. The decline is not uniform: homoglyph and spacing edits defeat a fixed lexicon most completely, because the altered string produces tokens the filter has never seen, whereas leetspeak is caught slightly more often when the filter applies basic character-substitution rules. This pattern reproduces the adversarial results of the comparison

studies almost exactly—Hosseini and colleagues drove a mature toxicity model's score from over 0.8 to near 0.2 with small perturbations [17], and Gröndahl and colleagues showed that inserting spaces or appending innocuous words defeated several published detectors [18]. The lesson for Quilt is that lexical matching alone is brittle: normalising text before filtering and adding character-level or learned matching would most directly close the evasion gap, and the human reporting queue remains the essential backstop for whatever slips through.

5. RELATED WORK

Maphosa developed the MyLSU application to enhance student engagement at a rural university, using a design-science and user-centered process to deliver off-campus features such as voting for student leaders and information on events and social activities; an evaluation with 220 participants found the application usable, easy to learn, and likely to be recommended [11]. More broadly, systematic reviews confirm that well-designed mobile tools can meaningfully increase engagement and interaction in educational settings [13]. Together these works establish that a mobile hub can strengthen a campus community, but they focus on engagement and usability alone and are aimed at university students. Quilt builds on the same premise while extending it in two directions: it targets the high-school setting, where scheduling, clubs, and parent involvement are central, and it treats safety as a first-class concern rather than assuming a self-governing adult audience.

A second line of work concerns applications that connect schools with families. Tools such as ClassDojo have been widely adopted to carry behavior updates and messages between teachers and parents, and controlled studies show that structured teacher-to-family communication can raise student engagement [15]. Critical analyses of these platforms, however, warn that they can drift toward surveillance and constant behavioral scoring, concentrating information in the hands of adults while giving students little agency [12]. These systems succeed at one-way, teacher-driven messaging but were not designed as a community that students themselves inhabit. Quilt differs by centering students as participants — they post, join clubs, and run polls — and by limiting the parent view to school-relevant activity established through a consent-based link, aiming to preserve the communication benefit without the surveillance dynamic these critiques identify.

A third body of research addresses the safety problem directly through automated content moderation. Surveys of the field describe pipelines built on natural-language processing and machine learning that detect toxic or harmful text for removal or review [10], and widely cited systems such as automated hate-speech classifiers demonstrate both the promise and the difficulty of the task, since offensive language is easily confused with merely coarse speech [9]. Scholars of moderation further note that no automated system is complete without human judgment and clear governance [8]. These approaches optimize detection in isolation, and they are typically studied on large open platforms rather than packaged for a specific community. Quilt deliberately combines a lightweight automated filter with human reporting inside a closed, identifiable school population, trading maximal classifier sophistication for a practical, layered safeguard suited to its scale.

6. CONCLUSIONS

Quilt has several limitations, most of which stem from the design decisions above rather than from measured performance, since a full evaluation of the system is still to be completed. The most important concerns moderation. The on-device content-safety filter relies on a fixed set of rules, which cannot anticipate every harmful phrasing and may both miss genuinely unsafe posts

and flag innocent ones; as the literature makes clear, this trade-off is inherent to automated detection [10][9]. Given more time, the filter could be replaced or supplemented with a trained classifier and evaluated against a labeled set of real posts, and the reporting queue could be studied to measure how quickly harmful content is removed in practice. The parent-connection feature depends on students and guardians completing an in-person QR exchange, which is secure but adds friction; a verified school-roster integration could make linking smoother without weakening privacy. The reliance on a single cloud provider for authentication, data, and storage also concentrates risk, so introducing redundancy and offline caching would improve resilience. Finally, because the community is scoped to each school, the application's value grows only once enough members of a school adopt it, so future work should address this onboarding threshold directly.

Quilt demonstrates that the scattered communication, organization, and community of a high school can be gathered into a single, school-scoped, and moderate application. Although its safeguards and features still require empirical evaluation, the overall design offers a practical template for making school life easier to follow, easier to participate in, and safer to share.

REFERENCES

[1] Anderson, Monica, Michelle Faverio, and Jeffrey Gottfried. "Teens, social media and technology 2023." Pew Research Center 11 (2023): 411-425.
[2] Rideout, Vicky. "The common sense census: Media use by tweens and teens." (2015).
[3] Fredricks, Jennifer A., and Jacquelynne S. Eccles. "Is extracurricular participation associated with beneficial outcomes? Concurrent and longitudinal relations." Developmental psychology 42.4 (2006): 698.
[4] Korpershoek, Hanke, et al. "The relationships between school belonging and students' motivational, social-emotional, behavioural, and academic outcomes in secondary education: A meta-analytic review." Research papers in education 35.6 (2020): 641-680.
[5] Jeynes, William H. "The relationship between parental involvement and urban secondary school student academic achievement: A meta-analysis." Urban education 42.1 (2007): 82-110.
[6] Selkie, Ellen M., Jessica L. Fales, and Megan A. Moreno. "Cyberbullying prevalence among US middle and high school–aged adolescents: A systematic review and quality assessment." Journal of Adolescent Health 58.2 (2016): 125-133.
[7] Hinduja, Sameer, and Justin W. Patchin. Bullying beyond the schoolyard: Preventing and responding to cyberbullying. Corwin press, 2014.
[8] Gillespie, Tarleton. Custodians of the Internet: Platforms, content moderation, and the hidden decisions that shape social media. Yale University Press, 2018.
[9] Davidson, Thomas, et al. "Automated hate speech detection and the problem of offensive language." Proceedings of the international AAAI conference on web and social media. Vol. 11. No. 1. 2017.
[10] Gongane, Vaishali U., Mousami V. Munot, and Alwin D. Anuse. "Detection and moderation of detrimental content on social media platforms: current status and future directions." Social network analysis and mining 12.1 (2022): 129.
[11] Maphosa, Vusumuzi. "Using MyLSU app to enhance student engagement and promote a smart town at a rural university in Zimbabwe." Cogent Education 7.1 (2020): 1823143.
[12] Williamson, Ben. "Decoding ClassDojo: Psycho-policy, social-emotional learning and persuasive educational technologies." Learning, Media and Technology 42.4 (2017): 440-453.
[13] Crompton, Helen, and Diane Burke. "The use of mobile learning in higher education: A systematic review." Computers & education 123 (2018): 53-64.
[14] Stellmann, Emily, and Liyan Song. "Essential elements of digital citizenship: Insights from the literature." TechTrends 68.4 (2024): 749-760.
[15] Kraft, Matthew A., and Shaun M. Dougherty. "The effect of teacher–family communication on student engagement: Evidence from a randomized field experiment." Journal of Research on Educational Effectiveness 6.3 (2013): 199-222.
[16] Wang, Ming-Te, and Jessica L. Degol. "School climate: A review of the construct, measurement, and impact on student outcomes." Educational psychology review 28.2 (2016): 315-352.

[17] Hosseini, Hossein, et al. "Deceiving google's perspective api built for detecting toxic comments." arXiv preprint arXiv:1702.08138 (2017).
[18] Gröndahl, Tommi, et al. "All you need is" love" evading hate speech detection." Proceedings of the 11th ACM workshop on artificial intelligence and security. 2018.
[19] Georgakopoulos, Spiros V., et al. "Convolutional neural networks for toxic comment classification." Proceedings of the 10th hellenic conference on artificial intelligence. 2018.

©2026 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.