**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**



**NHÓM 10**

**TÌM HIỂU VỀ APPSHEET – QUẢN LÝ NHÀ KHO**

**ĐỒ ÁN MÔN**

**CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI**

NGÀNH: CÔNG NGHỆ THÔNG TIN

**Thành phố Hồ Chí Minh, năm 2024**

**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**



**NHÓM 10**

**TÌM HIỂU VỀ APPSHEET – QUẢN LÝ NHÀ KHO**

**ĐỒ ÁN MÔN**

**CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI**

NGÀNH: CÔNG NGHỆ THÔNG TIN

**Giảng viên phụ trách**

**ThS. PHẠM THI VƯƠNG**

**Thành phố Hồ Chí Minh, năm 2024**

# **Lời cam đoan**

Em tên là Võ Đăng Quang – người viết báo cáo này, em đại diện nhóm 10 xin cam đoan rằng đồ án “*Tìm hiểu về Appsheet – Quản lý nhà kho*” là công trình nghiên cứu của nhóm 10 dưới sự hướng dẫn trực tiếp của ThS. Phạm Thi Vương.

Tất cả số liệu, thông tin, kết quả nghiên cứu, mọi trích dẫn sử dụng trong báo cáo đều được thu thập và thực hiện một cách trung thực, khách quan, ghi rõ nguồn tài liệu tham khảo theo đúng quy định.

Nhóm em xin hoàn toàn chịu trách nhiệm về nội dung và chịu mọi hình thức kỷ luật theo quy định nếu có bất kì hành vi vi phạm, gian trá nào.

**Thành phố Hồ Chí Minh, năm 2024**

**Nhóm 10**

# **Lời cảm ơn**

Lời đầu tiên nhóm em xin bày tỏ lời cảm ơn chân thành và sâu sắc nhất đến thầy ThS. Phạm Thi Vương, người trực tiếp hướng dẫn, góp ý và tận tình chỉ bảo cho em cho tới khi nhóm hoàn thành đồ án của mình.

Tiếp đến nhóm xin giành lời cảm ơn đến quý thầy cô Trường Đại học Sài Gòn – khoa Công nghệ thông tin đã truyền đạt cho em những kiến thức vô cùng quý báu và bổ ích trong suốt quá trình nghiên cứu và học tập tại trường.

Và cuối cùng, chân thành cảm ơn tới những thành viên trong nhóm đã luôn sát cánh cùng nhau, những lời động viên, những lần hỗ trợ những lúc cần thiết đã phần nào giúp cả nhóm hoàn thành đồ án này.

# **Tổng hợp**

Nhóm: 10

Tên đề tài: Tìm hiểu về Appsheet – Quản lý nhà kho

Thành viên nhóm:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và Tên** | **Khối lượng công việc** |
| 1 | 3120410429 | Võ Đăng Quang | 25% |
| 2 | 3120410437 | Nguyễn Văn Tấn Quân | 25% |
| 3 | 3120410428 | Triệu Khánh Quang | 25% |
| 4 | 3120410450 | Nguyễn Quốc Siêu | 25% |

Kế hoạch:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tuần** | **Thời gian** | **Công việc thực hiện** | **Ghi chú** |
| 1 | 15/1/2024 – 21/1/2024 | Thành lập nhóm 10 | Trao đổi thông qua Zalo |
| 2 | 22/1/2024 – 28/1/2024 | * Nhận gợi ý đề tài từ giảng viên hướng dẫn. * Nghiên cứu tên đề tài. |  |
| 3 | 19/2/2024 – 25/2/2024 | * Thống nhất đề tài. * Tìm hiêu các nền tảng khác có liên quan đến đề tài. * Nộp đề cương chi tiết. | Trao đổi thông qua Zalo, Discord và Github |
| 4 | 26/2/2024 – 3/3/2024 |
| 5 | 4/3/2024 – 10/3/2024 | * Nghiên cứu các chức năng của Appsheet * Trao đổi với giảng viên các vấn đề khó khăn gặp phải | Các vấn đề đã được giải quyết sau khi có được sự gợi ý từ giảng viên |
| 6 | 11/3/2024 – 17/3/2024 |
| 7 | 18/3/2024 – 24/3/2024 | * Nghiên cứu các chức năng cơ bản của Appsheet * Hoàn tất chương 1 | Bắt đầu viết báo cáo |
| 8 | 25/3/2024 – 31/3/2024 | * Khởi động dự án * Hoàn tất chương 2 và 3 |  |
| 9 | 1/4/2024 – 7/4/2024 |  |
| 10 | 8/4/2024 – 14/4/2024 |  |
| 11 | 15/4/2024 – 21/4/2024 |  |
| 12 | 22/4/2024 – 28/4/2024 | * Kiểm thử và kết thúc đồ án | Hoàn tất báo cáo |
| 13 | 29/4/2024 – 5/5/2024 | Báo cáo đồ án theo lịch mà giảng viên hướng dẫn đã đưa ra trước đó |  |
| 14 | 6/5/2024 – 12/5/2024 |  |
| 15 | 13/5/2024 – 19/5/2024 |  |

# **Mục lục**

[**Lời cam đoan** i](#_Toc164938266)

[**Lời cảm ơn** ii](#_Toc164938267)

[**Tổng hợp** iii](#_Toc164938268)

[**Mục lục** v](#_Toc164938269)

[**Danh mục hình ảnh** viii](#_Toc164938270)

[**Danh mục bảng biểu** xi](#_Toc164938271)

[**Lời mở đầu** 1](#_Toc164938272)

[**Chương 1: TỔNG QUAN VỀ APPSHEET** 3](#_Toc164938273)

[**1.1.** **Giới thiệu về App No-code** 3](#_Toc164938274)

[**1.2.** **Lợi ích của việc sử dụng App No-code** 4](#_Toc164938275)

[**1.3.** **Ứng dụng trong công việc hiệu quả** 4](#_Toc164938276)

[**1.3.1.** **Dưới góc nhìn của lập trình viên** 4](#_Toc164938277)

[**1.3.2.** **Dưới góc nhìn của doanh nghiệp** 5](#_Toc164938278)

[**1.4.** **Ví dụ về nền tảng no-code phổ biến:** 5](#_Toc164938279)

[**1.5.** **Tổng quan về Appsheet** 6](#_Toc164938280)

[**1.5.1.** **Ưu điểm** 7](#_Toc164938281)

[**1.5.2.** **Nhược điểm** 7](#_Toc164938282)

[**1.6.** **Cách Appsheet hoạt động** 8](#_Toc164938283)

[**1.6.1.** **Dữ liệu** 8](#_Toc164938284)

[**1.6.2.** **Giao diện** 8](#_Toc164938285)

[**1.6.3.** **Chức năng** 9](#_Toc164938286)

[**1.6.4.** **Triển khai** 9](#_Toc164938287)

[**1.7.** **Sử dụng Appsheet trong doanh nghiệp** 10](#_Toc164938288)

[**Chương 2: NGHIÊN CỨU VỀ APPSHEET** 13](#_Toc164938289)

[**2.1.** **Giới hạn của Appsheet** 13](#_Toc164938290)

[**2.2.** **Dữ liệu và cách làm việc với dữ liệu trên Appsheet** 14](#_Toc164938291)

[**2.2.1.** **Kết nối dữ liệu** 14](#_Toc164938292)

[**2.2.2.** **Định dạng dữ liệu** 15](#_Toc164938293)

[**2.3.** **View** 16](#_Toc164938294)

[**2.4.** **Action** 18](#_Toc164938295)

[**2.5.** **Automation** 20](#_Toc164938296)

[**2.6.** **Các chức năng nâng cao trong Appsheet** 21](#_Toc164938297)

[**2.6.1.** **Tính năng nâng cao về dữ liệu:** 21](#_Toc164938298)

[**2.6.2.** **Tính năng nâng cao về ứng dụng:** 22](#_Toc164938299)

[**2.6.3.** **Tính năng quản lý và cộng tác:** 22](#_Toc164938300)

[**Chương 3: ỨNG DỤNG APPSHEET ĐỂ XÂY DỰNG ỨNG DỤNG QUẢN LÝ KHO HÀNG** 23](#_Toc164938301)

[**3.1.** **Đặc tả** 23](#_Toc164938302)

[**3.1.1.** **Mô tả ứng dụng** 23](#_Toc164938303)

[**3.1.1.1.** **Quản lý người dùng** 24](#_Toc164938304)

[**3.1.1.2.** **Quản lý danh sách kho** 24](#_Toc164938305)

[**3.1.1.3.** **Quản lý sản phẩm** 25](#_Toc164938306)

[**3.1.2.** **Các chức năng ứng dụng** 26](#_Toc164938307)

[**3.1.2.1.** **Quản lý nhập xuất** 26](#_Toc164938308)

[**3.1.2.2.** **Điều chỉnh, kiểm kê** 27](#_Toc164938309)

[**3.1.2.3.** **Báo cáo nhập – xuất – tồn** 27](#_Toc164938310)

[**3.2.** **Tạo cơ sở dữ liệu** 28](#_Toc164938311)

[**3.2.1.** **Bảng dữ liệu User** 28](#_Toc164938312)

[**3.2.2.** **Bảng dữ liệu Phieu NX** 29](#_Toc164938313)

[**3.2.3.** **Bảng dữ liệu PhieuNXChitiet** 30](#_Toc164938314)

[**3.2.4.** **Bảng dữ liệu DSKho** 30](#_Toc164938315)

[**3.2.5.** **Bảng dữ liệu SanPham** 31](#_Toc164938316)

[**3.3.** **Tạo ứng dụng** 31](#_Toc164938317)

[**3.4.** **Giao diện quản lý thông tin** 36](#_Toc164938318)

[**3.4.1.** **Các thiết lập và ràng buộc dữ liệu** 36](#_Toc164938319)

[**3.4.1.1.** **Điều chỉnh bảng User** 38](#_Toc164938320)

[**3.4.1.2.** **Điều chỉnh bảng Phieu NX** 40](#_Toc164938321)

[**3.4.1.3.** **Điều chỉnh bảng PhieuNXChitiet** 46](#_Toc164938322)

[**3.4.1.4.** **Điều chình bảng DSKho** 47](#_Toc164938323)

[**3.4.1.5.** **Điều chỉnh bảng SanPham** 48](#_Toc164938324)

[**3.4.2.** **Dùng view để tạo giao diện cho ứng dụng** 48](#_Toc164938325)

[**3.5.** **Chức năng quản lý nhập xuất** 49](#_Toc164938326)

[**3.5.1.** **Chức năng nhập kho** 49](#_Toc164938327)

[**3.5.2.** **Chức năng xuất kho** 49](#_Toc164938328)

[**3.5.3.** **Xuất phiếu nhập xuất kho** 49](#_Toc164938329)

[**3.6.** **Tổng hợp dữ liệu để kiểm kê** 49](#_Toc164938330)

[**3.6.1.** **Tổng hợp dữ liệu** 49](#_Toc164938331)

[**3.6.2.** **Tổng hợp dữ liệu của từng kho** 49](#_Toc164938332)

[**3.6.3.** **Kiểm kê** 49](#_Toc164938333)

[**3.7.** **Backup dữ liệu** 49](#_Toc164938334)

[**3.8.** **Demo ứng dụng** 49](#_Toc164938335)

[**Kết luận và hướng phát triển** 50](#_Toc164938336)

[**Kết luận** 50](#_Toc164938337)

[**Hướng phát triển** 50](#_Toc164938338)

[**Tài liệu tham thảo** 51](#_Toc164938339)

# **Danh mục hình ảnh**

[Hình 1.1. Một số nền tảng tạo ứng dụng không cần code phổ biến hiện nay 1](#_Toc164718347)

[Hình 1.2. App là gì 3](#_Toc164718348)

[Hình 1.3. Một số lợi ích của App No-code so với App truyền thống 4](#_Toc164718349)

[Hình 1.4. Praveen Seshadri (trái) và Brian Sabino (phải), đồng sáng lập Appsheet 6](#_Toc164718350)

[Hình 1.5. Cách Appsheet hoạt động 9](#_Toc164718351)

[Hình 1.6. AppSheet trong doanh nghiệp 10](#_Toc164718352)

[Hình 2.1. Dữ liệu được lưu trữ bằng Google Sheets 14](#_Toc164938217)

[Hình 2.2. Tên views 17](#_Toc164938218)

[Hình 2.3. View options 17](#_Toc164938219)

[Hình 2.4. Icon minus đối với các views không cần cho người nhìn thấy 17](#_Toc164938220)

[Hình 2.5. Icon hiển thị đồng nhất 18](#_Toc164938221)

[Hình 2.6. Sử dụng màu trong view 18](#_Toc164938222)

[Hình 2.7. Đánh số thứ tự cho các action 20](#_Toc164938223)

[Hình 3.1. Đặc tả ứng dụng 23](#_Toc164962060)

[Hình 3.2. Bảng dữ liệu người dùng 24](#_Toc164962061)

[Hình 3.3. Bảng dữ liệu danh sách kho 25](#_Toc164962062)

[Hình 3.4. Bảng dữ liệu sản phẩm 25](#_Toc164962063)

[Hình 3.5. Chức năng nhập xuất của ứng dụng 26](#_Toc164962064)

[Hình 3.6. Phiếu nhập kho 26](#_Toc164962065)

[Hình 3.7. Trực quan hoá báo cáo nhập - xuất - tồn 27](#_Toc164962066)

[Hình 3.8. Chức năng xuất ra file Excel 27](#_Toc164962067)

[Hình 3.9. File báo cáo Excel 28](#_Toc164962068)

[Hình 3.10. Các bảng dữ liệu trong ứng dụng 28](#_Toc164962069)

[Hình 3.11. Bảng dữ liệu User 28](#_Toc164962070)

[Hình 3.12. Bảng dữ liệu Phieu NX 29](#_Toc164962071)

[Hình 3.13. Bảng dữ liệu PhieuNXChitiet 30](#_Toc164962072)

[Hình 3.14. Bảng dữ liệu DSKho 30](#_Toc164962073)

[Hình 3.15. Bảng dữ liệu SanPham 31](#_Toc164962074)

[Hình 3.16. Trang chủ Appsheet 32](#_Toc164962075)

[Hình 3.17. Tạo 1 App mới 32](#_Toc164962076)

[Hình 3.18. Đặt tên cho App 32](#_Toc164962077)

[Hình 3.19. Chọn loại cơ sở dữ liệu 33](#_Toc164962078)

[Hình 3.20. Chọn cơ sở dữ liệu đã tạo sẵn 33](#_Toc164962079)

[Hình 3.21. App được tạo thành công 34](#_Toc164962080)

[Hình 3.22. Giao diện vertical mobile 35](#_Toc164962081)

[Hình 3.23. Giao diện horizontal mobile 36](#_Toc164962082)

[Hình 3.24. Giao diện desktop 36](#_Toc164962083)

[Hình 3.25. Cơ sở dữ liệu chỉ có mỗi bảng User 37](#_Toc164962084)

[Hình 3.26. Thêm cấc bảng dữ liệu khác vào cơ sở dữ liệu 37](#_Toc164962085)

[Hình 3.27. Cơ sở dữ liệu đã được thêm đầy đủ 38](#_Toc164962086)

[Hình 3.28. Các trường dữ liệu trong bảng User 38](#_Toc164962087)

[Hình 3.29. Ràng buộc trong bảng User 39](#_Toc164962088)

[Hình 3.30. Các trường dữ liệu trong bảng Phieu NX 40](#_Toc164962089)

[Hình 3.31. 2 Kiểu dữ liệu trong Loại NX 41](#_Toc164962090)

[Hình 3.32. Kho sẽ lấy dữ liệu từ bảng DSKho 41](#_Toc164962091)

[Hình 3.33. Thủ kho sẽ lấy dữ liệu từ bảng User 42](#_Toc164962092)

[Hình 3.34. Cấc kiểu dữ liệu trong Diễn giải 43](#_Toc164962093)

[Hình 3.35. Ràng buộc trong bảng Phieu NX 44](#_Toc164962094)

[Hình 3.36. Công thức không cho phép người dùng chỉnh sửa dữ liệu 44](#_Toc164962095)

[Hình 3.37. Công thức tự động đặt tên phiếu nhập xuất 44](#_Toc164962096)

[Hình 3.38. Hàm TODAY() 45](#_Toc164962097)

[Hình 3.39. Công thức phân biệt loại xuất nhập 45](#_Toc164962098)

[Hình 3.40. Công thức phân biệt các lựa chọn diễn giải cho từng loại phiếu 45](#_Toc164962099)

[Hình 3.41. Công thức dùng API của bên thứ 3 để tạo QR Code 45](#_Toc164962100)

[Hình 3.42. Kết quả khi quét QR Code trên điện thoại 46](#_Toc164962101)

[Hình 3.43. Các trường dữ liệu trong bảng PhieuNXChitiet 47](#_Toc164962102)

[Hình 3.44. Chọn Is a part of? 48](#_Toc164962103)

[Hình 3.45. Sản phẩm lấy dữ liệu từ bảng SanPham 48](#_Toc164962104)

[Hình 3.46. Nhập điều kiện để có thể đọc dữ liệu từ bảng SanPham 49](#_Toc164962105)

[Hình 3.47. Ràng buộc trong bảng PhieuNXChitiet 49](#_Toc164962106)

[Hình 3.48. Tham chiếu đến cột Tên Sản Phẩm trong bảng SanPham 50](#_Toc164962107)

[Hình 3.49. Tham chiếu đến cột Mô tả trong bảng SanPham 50](#_Toc164962108)

[Hình 3.50. Tham chiếu đến cột Đơn vị tính trong bảng SanPham 50](#_Toc164962109)

[Hình 3.51. Công thức SL NX 50](#_Toc164962110)

[Hình 3.52. Các trường dữ liệu trong bảng DSKho 50](#_Toc164962111)

[Hình 3.53. Ràng buộc trong bảng DSKho 51](#_Toc164962112)

[Hình 3.54. Các trường dữ liệu trong bảng SanPham 51](#_Toc164962113)

[Hình 3.55. Ràng buộc trong bảng SanPham 52](#_Toc164962114)

[Hình 3.56. Vào Views 52](#_Toc164962115)

[Hình 3.57. Nơi chức các chức năng của ứng dụng 53](#_Toc164962116)

[Hình 3.58. Thêm Views mới 53](#_Toc164962117)

[Hình 3.59. Danh sách các Views sẽ được chỉnh sửa 54](#_Toc164962118)

[Hình 3.60. Views của User 54](#_Toc164962119)

[Hình 3.61. Chọn icon cho User 55](#_Toc164962120)

[Hình 3.62. Views của Danh sách kho 55](#_Toc164962121)

[Hình 3.63. Chọn icon cho Danh sách kho 56](#_Toc164962122)

[Hình 3.64. Views của Sản Phẩm 56](#_Toc164962123)

[Hình 3.65. Chọn icon cho Sản Phẩm 57](#_Toc164962124)

[Hình 3.66. Tạo Slice Phiếu Nhập 57](#_Toc164962125)

[Hình 3.67. Tạo 1 Slice mới 57](#_Toc164962126)

[Hình 3.68. Chỉnh sửa thông tin slice Phiếu Nhập 58](#_Toc164962127)

[Hình 3.69. Chọn Create a custom expression 58](#_Toc164962128)

[Hình 3.70. Công thức trong slice Phiếu Nhập 58](#_Toc164962129)

[Hình 3.71. Views của Phiếu Nhập 59](#_Toc164962130)

[Hình 3.72. Chọn icon cho Phiếu Nhập 59](#_Toc164962131)

[Hình 3.73. Duplicate Phiếu Nhập để tạo nhanh Phiếu Xuất 60](#_Toc164962132)

[Hình 3.74. Chỉnh sửa thông tin của Slide: Phiếu Xuất 60](#_Toc164962133)

[Hình 3.75. Views của Phiếu Xuất 61](#_Toc164962134)

[Hình 3.76. Chọn icon cho Phiếu Xuất 61](#_Toc164962135)

[Hình 3.77. Giao diện ứng dụng với đầy đủ chức năng 62](#_Toc164962136)

[Hình 3.78. Actions 62](#_Toc164962137)

[Hình 3.79. Danh sách các dữ liệu có thể tạo Actions 62](#_Toc164962138)

[Hình 3.80. Tạo action mới 63](#_Toc164962139)

[Hình 3.81. Action Tạo đường dẫn file Phiếu NX 63](#_Toc164962140)

[Hình 3.82. Công thức tạo đường dẫn file PDF 64](#_Toc164962141)

[Hình 3.83. Chọn icon cho action tạo đường dẫn file PDF 64](#_Toc164962142)

[Hình 3.84. Action Sync 65](#_Toc164962143)

[Hình 3.85. Công thức đồng bộ ứng dụng 65](#_Toc164962144)

[Hình 3.86. Action Mở phiếu NX 66](#_Toc164962145)

[Hình 3.87. Chọn icon cho action Mở phiếu NX 66](#_Toc164962146)

[Hình 3.88. Action Tạo phiếu NX 67](#_Toc164962147)

[Hình 3.89. Chọn icon cho action Tạo phiếu NX 67](#_Toc164962148)

[Hình 3.90. Action Open File 68](#_Toc164962149)

[Hình 3.91. Automation 68](#_Toc164962150)

[Hình 3.92. Tạo bot mới 68](#_Toc164962151)

[Hình 3.93. Chỉnh sửa event cho bot 69](#_Toc164962152)

[Hình 3.94. Công thức bắt sự kiện 70](#_Toc164962153)

[Hình 3.95. Add a step 70](#_Toc164962154)

[Hình 3.96. Chỉnh sửa step của bot (1) 71](#_Toc164962155)

[Hình 3.97. Chỉnh sửa step của bot (2) 72](#_Toc164962156)

[Hình 3.98. Công thức đặt tên file PDF 73](#_Toc164962157)

# **Danh mục bảng biểu**

[Bảng 1.1. Các doanh nghiệp ba mức quy mô sử dụng Appsheet 11](#_Toc164718332)

[Bảng 2.1. Bảng so sánh giới hạn của 2 phiên bản appsheet 14](#_Toc164718339)

# **Lời mở đầu**

Xu hướng tạo ứng dụng di động và web không cần code đang ngày càng phát triển mạnh mẽ. Nhu cầu xây dựng ứng dụng ngày càng cao, tuy nhiên không phải ai cũng có khả năng lập trình. Do đó, các nền tảng tạo ứng dụng không cần code ra đời như một giải pháp hữu hiệu.





*Hình 1.1. Một số nền tảng tạo ứng dụng không cần code phổ biến hiện nay*

Hiện nay có rất nhiều nền tảng tạo ứng dụng không cần code phổ biến như: AppSheet, Zoho Creator, Airtable, Google App Maker, Microsoft Power Apps, Mendix, OutSystems,... Mỗi nền tảng có ưu và nhược điểm riêng, phù hợp với nhu cầu sử dụng khác nhau. Trong đồ án này nhóm em chọn AppSheet làm nền tảng nghiên cứu chính.

AppSheet là một nền tảng cho phép bạn xây dựng các ứng dụng di động và web tùy chỉnh mà không cần viết mã. Nó sử dụng giao diện kéo, thả đơn giản để bạn có thể tạo các ứng dụng dựa trên dữ liệu trong Google Sheets, Google Drive và các nguồn dữ liệu khác có sẵn miễn phí cho các ứng dụng cá nhân. Ngoài ra, AppSheet còn có một số ưu điểm khác so với các nền tảng khác:

* Nhanh chóng: AppSheet cho phép bạn tạo ứng dụng nhanh chóng hơn so với các nền tảng khác.
* Dễ học: AppSheet dễ học hơn so với các nền tảng khác.
* Hỗ trợ tốt: AppSheet có cộng đồng hỗ trợ mạnh mẽ và đội ngũ hỗ trợ khách hàng chuyên nghiệp.

Những nghiên cứu về tính hiệu quả của Appsheet sẽ được giải thích vì sao các doanh nghiệp áp dụng nó cho hệ thống của mình. Vì vậy, để tìm hiểu rõ hơn về cách hoạt động của Appsheet cũng như cách các doanh nghiệp sẽ ứng dụng nó, nhóm em quyết định chọn nghiên cứu đề tài: “Tìm hiểu về Appsheet – Quản lý nhà kho”.

Trong đề tài này, nhóm sẽ trình bày rõ thế nào là một phần mềm được tạo từ Appsheet. Kết hợp với việc quan sát trực quan trên các video hướng dẫn và các bài tập đã làm, nhóm sẽ cho ra các nhận định của mình. Và cuối cùng, nhóm em sử dụng phương pháp thực nghiệm để xây dựng một phần mềm bằng Appsheet.

Đồ án của nhóm 10 được chia thành các phần như sau:

Chương 1: Trình bày tổng quan về khái niệm Appsheet. Đồng thời, nhóm em cũng trình bày rõ ưu và nhược điểm của nó.

Chương 2: Trong chương này, nhóm sẽ trình bày các nghiên cứu của nhóm về Appsheet.

Chương 3: Xây dựng phần mềm từ khâu đặc tả cho tới chạy thử và đưa ra kết luận cũng như hướng phát triển cho phần mềm.

# **Chương 1: TỔNG QUAN VỀ APPSHEET**

## **Giới thiệu về App No-code**

Trước khi tìm hiểu App No-code là gì ta phải hiểu app là gì trước đã. Nói một cách đơn giản App (Ứng dụng) là một chương trình phần mềm được thiết kế để thực hiện các tác vụ cụ thể trên thiết bị điện tử như smartphone, tablet, laptop,…



*Hình 1.2. App là gì*

App có thể được cài đặt từ các kho ứng dụng như App Store (iOS), Google Play Store (Android), Microsoft Store (Windows) hoặc tải trực tiếp từ trang web của nhà phát triển.

Một App có thể có nhiều mục đích sử dụng khác nhau. Và để phát triển các ứng dụng, ta phải code. Để làm được điều đó ta cần các lâp trình viên tham gia quá trình xây dựng, phát triển. Một App xuất hiện trên các kho ứng dụng cần rất nhiều thời gian, công sức và tiền bạc để phát triển. Vậy thì phát triển một App No-code nghĩa là như thế nào?

Nền tảng App No-Code (hay còn gọi là nền tảng phát triển ứng dụng không cần viết mã) là một hệ thống cho phép người dùng tạo ứng dụng di động, web và máy tính mà không cần có kiến thức lập trình chuyên sâu. Thay vì viết mã truyền thống, người dùng sử dụng các công cụ trực quan như kéo thả, chọn lựa và cấu hình để xây dựng giao diện, logic và chức năng cho ứng dụng.

## **Lợi ích của việc sử dụng App No-code**



*Hình 1.3. Một số lợi ích của App No-code so với App truyền thống*

* Tiết kiệm thời gian và chi phí: Việc phát triển ứng dụng no-code diễn ra nhanh chóng hơn nhiều so với lập trình truyền thống, giúp giảm thiểu chi phí cho nhân sự và nguồn lực.
* Dễ sử dụng: Giao diện trực quan và các công cụ đơn giản giúp mọi người, bất kể trình độ kỹ thuật, đều có thể tạo ứng dụng.
* Tính linh hoạt: Nền tảng No-code cung cấp nhiều tính năng và khả năng tùy chỉnh, đáp ứng nhu cầu đa dạng của người dùng.
* Khả năng mở rộng: App No-code có thể dễ dàng mở rộng và nâng cấp để đáp ứng nhu cầu phát triển trong tương lai.

## **Ứng dụng trong công việc hiệu quả**

### **Dưới góc nhìn của lập trình viên**

* Phải có code.
* Phải làm chủ nền tảng.
* Dữ liệu phải lưu local.
* Phải nắm được source của phần mềm.

### **Dưới góc nhìn của doanh nghiệp**

* Mọi phần mềm, app chỉ đơn giản là công cụ.
* Đối với doanh nghiệp để chọn một nền tàng thích hợp cần phân tích nhu cầu, ngân sách và hiệu quả mang lại. Các tiêu chí được ưu tiên như sau:
* Hiệu quả
* Bảo mật
* Hỗ trợ lâu dài
* Chi phí hợp lý
* Giải quyết được vấn đề cho doanh nghiệp
* Dễ dàng kết nối với các hệ sinh thái trong tương lai

Suy cho cùng, No-code hay code cũng chỉ là công cụ. Mục đích cuối cùng là tối ưu hoá công việc. Để áp dụng App No-code vào công việc hiệu quả, chúng ta cần nắm vững các yếu tố sau:

* Nắm rõ mục tiêu chính trong quá trình tạo App, loại bỏ các phần gây xao nhãng.
* Biết và hiểu về nền tảng, điểm mạnh, điểm hạn chế.
* Xử lý các vấn đề tồn đọng hiện tại của doanh nghiệp.
* Chi phí phù hợp với yêu cầu doanh nghiệp.
* Định hướng phát triển trong tương lai.

## **Ví dụ về nền tảng no-code phổ biến:**

* AppSheet: Nền tảng no-code mạnh mẽ cho phép kết nối với nhiều nguồn dữ liệu và tạo ứng dụng di động đa nền tảng.
* Adalo: Nền tảng no-code thân thiện với người dùng, cung cấp các công cụ trực quan để xây dựng ứng dụng di động đẹp mắt.
* Bubble: Nền tảng no-code linh hoạt, cho phép tạo ứng dụng web và di động phức tạp với nhiều tính năng.
* Glide: Nền tảng no-code tập trung vào việc tạo ứng dụng từ bảng tính Google Sheets, Airtable và các nguồn dữ liệu khác.

Các nền tảng App No-Code là một công cụ mạnh mẽ và hữu ích giúp mọi người dễ dàng tạo ứng dụng mà không cần kiến thức lập trình chuyên sâu. Với sự phát triển không ngừng của công nghệ no-code, ngày càng nhiều người sẽ có thể tiếp cận và sử dụng nền tảng này để hiện thực hóa ý tưởng sáng tạo của mình.

## **Tổng quan về Appsheet**

Appsheet được phát triển bởi công ty AppSheet Inc. được thành lập vào năm 2014 bởi Praveen Seshadri và Brian Sabino.



*Hình 1.4. Praveen Seshadri (trái) và Brian Sabino (phải), đồng sáng lập Appsheet*

Năm 2020, Appsheet được mua lại bởi Google và trở thành một phần của Google Cloud Platform.

Đây là nền tảng tự động hoá giúp bạn tạo mobile app hay web một cách nhanh chóng mà không cần code. Nền tảng này đơn giản, trực quan và dễ sử dụng, cho phép chúng ta dễ dàng biến ý tưởng thành hiện thực chỉ với vài thao tác đơn giản. Với AppSheet, không cần có kinh nghiệm lập trình để tạo và quản lý các ứng dụng. Để sử dụng AppSheet, chỉ cần có kiến thức cơ bản về bảng tính hoặc cơ sở dữ liệu và có thể sử dụng các công cụ kéo và thả để tạo ra ứng dụng hoàn chỉnh. Tuy nhiên, ta nên cân nhắc nhu cầu của mình và biết những hạn chế của AppSheet trước khi quyết định sử dụng nó.

### **Ưu điểm**

* Dễ sử dụng: Giao diện kéo thả trực quan, dễ dàng sử dụng cho mọi người, kể cả những người không có kinh nghiệm lập trình.
* Nhanh chóng: Tạo ứng dụng nhanh chóng chỉ trong vài phút thay vì vài tháng.
* Tiết kiệm chi phí: Giảm thiểu chi phí phát triển ứng dụng đáng kể so với phương pháp truyền thống.
* Mạnh mẽ: Tạo ứng dụng với đầy đủ chức năng và khả năng tùy chỉnh cao.
* Kết nối dữ liệu: Kết nối với nhiều nguồn cở sở dữ liệu khác nhau như Google Sheet, Google Drive, Salesforce, SQL Server,…
* Đa dạng ứng dụng: Tạo ứng dụng cho mọi nhu cầu, từ quản lý công việc, kinh doanh đến theo dõi sức khỏe, giáo dục,…
* Miễn phí cho các ứng dụng cá nhân: AppSheet có sẵn miễn phí cho các ứng dụng cá nhân.

### **Nhược điểm**

* So với các nền tảng phát triển ứng dụng truyền thống, AppSheet vẫn còn một số thiếu sót như:
* Xử lý dữ liệu phức tạp.
* Tích hợp với các hệ thống bên ngoài.
* Quản lý người dùng và quyền truy cập.
* Tùy chỉnh giao diện và chức năng nâng cao hơn.
* Khả năng mở rộng: với các ứng dụng quy mô lớn, khả năng mở rộng dữ liệu, chức năng và hiệu suất có thể bị giới hạn.
* Khả năng tùy chỉnh:
* Mặc dù có khả năng tùy chỉnh cao, nhưng AppSheet vẫn có thể hạn chế trong việc tạo giao diện và chức năng độc đáo.
* Giao diện kéo thả có thể không đáp ứng được nhu cầu thiết kế phức tạp.
* Hỗ trợ: Hỗ trợ khách hàng của AppSheet có thể hạn chế so với các nền tảng trả phí. Thời gian chờ đợi hỗ trợ có thể lâu hơn và tài nguyên hỗ trợ có thể ít hơn.

Tuy nhiên, AppSheet là một nền tảng vừa được phổ biến gần đây nên không thể tránh khỏi việc có các nhược điểm nêu trên. Nhờ cộng đồng ngày một tăng và liên tục được cải tiến, cập nhật thêm tính năng mới, Appsheet sẽ trở thành một công cụ đắc lực cho doanh nghiệp trong tương lai.

## **Cách Appsheet hoạt động**

AppSheet là một nền tảng đột phá cho phép bạn tạo ứng dụng di động và web mạnh mẽ mà không cần viết mã. Nền tảng sử dụng giao diện kéo thả trực quan, giúp bạn dễ dàng biến ý tưởng thành hiện thực chỉ với vài thao tác đơn giản.

Với AppSheet, bạn có thể kết nối với nhiều nguồn dữ liệu khác nhau như Google Trang tính, Google Drive, Salesforce, SQL Server, v.v. và tạo ứng dụng cho mọi nhu cầu, từ quản lý công việc, kinh doanh đến theo dõi sức khỏe, giáo dục,…

### **Dữ liệu**

Biểu tượng bảng tính đại diện cho nguồn dữ liệu của bạn, chẳng hạn như Google Trang tính, Google Drive, Salesforce, SQL Server, v.v. AppSheet kết nối với nguồn dữ liệu của bạn và truy xuất dữ liệu để sử dụng trong ứng dụng.

### **Giao diện**

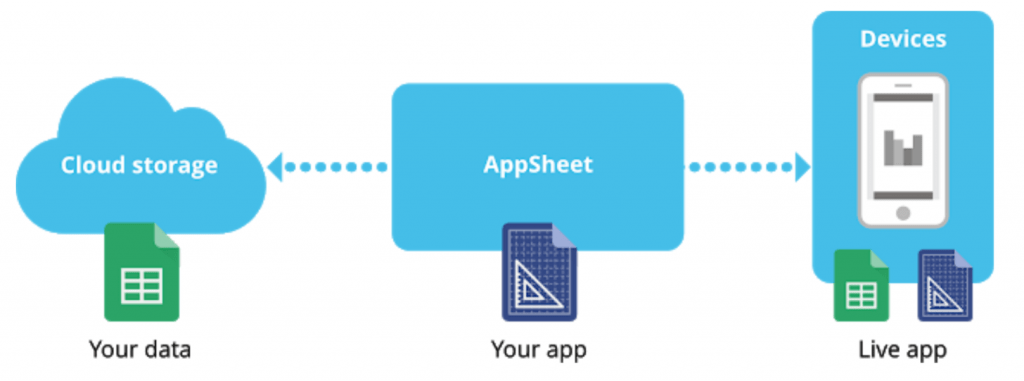
Biểu tượng điện thoại và máy tính đại diện cho giao diện người dùng của ứng dụng di động và web. Bạn sử dụng giao diện kéo thả trực quan để thiết kế giao diện người dùng cho ứng dụng của mình.

### **Chức năng**

Biểu tượng bánh răng đại diện cho logic có hướng dữ liệu được sử dụng để xác định chức năng cho ứng dụng của bạn. AppSheet cung cấp các công cụ để tạo quy tắc, xử lý sự kiện và tự động hóa các tác vụ.

### **Triển khai**

Biểu tượng AppSheet Marketplace đại diện cho khả năng chia sẻ ứng dụng của bạn với người dùng khác hoặc công bố lên AppSheet Marketplace. Bạn có thể quản lý và cập nhật ứng dụng của bạn sau khi triển khai.



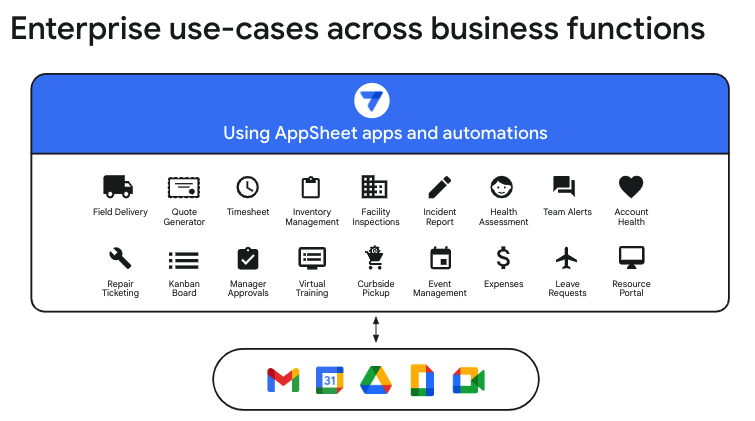
*Hình 1.5. Cách Appsheet hoạt động*

Trong hình 1.3, Dữ liệu được lưu trữ trong đám mây, sau đó AppSheet xử lý để tạo ra ứng dụng và cuối cùng, ứng dụng có thể được sử dụng trên các thiết bị di động và web. Hình ảnh cũng cho thấy các bước chính trong quá trình tạo ứng dụng với AppSheet:

* Kết nối dữ liệu: Kết nối AppSheet với nguồn dữ liệu của bạn.
* Thiết kế giao diện: Sử dụng giao diện kéo thả để thiết kế giao diện người dùng cho ứng dụng của bạn.
* Xác định chức năng: Sử dụng logic có hướng dữ liệu để xác định chức năng cho ứng dụng của bạn.
* Kiểm tra và triển khai: Kiểm tra ứng dụng của bạn và triển khai cho người dùng.

Nhìn chung, hình ảnh cung cấp một cái nhìn tổng quan về cách thức hoạt động của AppSheet và các bước chính trong quá trình tạo ứng dụng.

## **Sử dụng Appsheet trong doanh nghiệp**



*Hình 1.6. AppSheet trong doanh nghiệp*

AppSheet là một nền tảng mạnh mẽ, với nó, người dùng có thể thu thập dữ liệu phong phú thông qua biểu mẫu, mã vạch, địa điểm, chữ ký, và hình ảnh, đồng thời tùy chỉnh các tính năng, trải nghiệm người dùng, bảo mật,… Nền tảng này cũng cung cấp khả năng quản lý người dùng với bảo mật dựa trên vai trò, xác thực, và báo cáo sử dụng.

AppSheet phù hợp với mọi quy mô doanh nghiệp. Với các doanh nghiệp nhỏ, tạo ứng dụng đơn giản để quản lý vừa tiết kiệm được chi phí thuê kĩ thuật viên lập trình lại vừa tăng hiệu quả hoạt động. Còn đối với doanh nghiệp vừa và lớn, để tối ưu hóa hiệu quả và mở rộng thị trường thì ta cần thiết kế một ứng dụng phức tạp hơn cho các quy trình kinh doanh chuyên biệt nhưng vẫn đảm bảo về mặt chi phí. Đây cũng không phải điều bất khả thi kể cả khi Appsheet còn nhiều hạn chế khi thiết kế một ứng dụng phức tạp nên các doanh nghiệp vừa và lớn vẫn cứ yên tâm sử dụng Appsheet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quy mô doanh nghiệp | Nhu cầu | Ứng dụng | Lợi ích | Ví dụ |
| Nhỏ | Quy trình đơn giản, nguồn lực hạn chế | Quản lý dự án, theo dõi khách hàng tiềm năng, quản lý chi tiêu | Tiết kiệm chi phí, tăng hiệu quả | Quán cà phê: Quản lý đơn hàng, theo dõi nguyên liệu, thu ngân  Cửa hàng bán lẻ: Quản lý kho hàng, theo dõi doanh thu, cập nhật giá cả |
| Vừa | Quy trình phức tạp hơn, nhiều bộ phận tham gia | Quản lý quy trình bán hàng, quản lý quan hệ khách hàng (CRM), quản lý sản xuất | Tăng năng suất, cải thiện dịch vụ khách hàng, tối ưu hóa quy trình | Doanh nghiệp bán lẻ: Quản lý đơn hàng trực tuyến, theo dõi hàng tồn kho, hỗ trợ khách hàng trực tuyến  Công ty dịch vụ: Quản lý lịch trình nhân viên, theo dõi tiến độ công việc, báo cáo khách hàng |
| Lớn | Quy trình phức tạp, nhiều dữ liệu, nhiều người dùng | Tích hợp hệ thống ERP, tự động hóa quy trình, phân tích dữ liệu | Tăng hiệu quả hoạt động, giảm rủi ro, đưa ra quyết định sáng suốt | Doanh nghiệp sản xuất: Quản lý sản xuất, theo dõi chất lượng sản phẩm, giám sát dây chuyền sản xuất  Công ty tài chính: Phân tích dữ liệu tài chính, quản lý rủi ro, báo cáo cho ban lãnh đạo |

*Bảng 1.1. Các doanh nghiệp ba mức quy mô sử dụng Appsheet*

AppSheet giúp tự động hóa các quy trình làm việc, từ việc gửi và tự động hóa email, SMS, và thông báo đẩy. Điều này giúp doanh nghiệp tập trung vào công việc quan trọng hơn và giảm thiểu công việc thủ công. Ngoài ra, AppSheet có khả năng tích hợp mạnh mẽ với Google Workspace và các ứng dụng của bên thứ ba như Office 365, Dropbox, và Salesforce.com, cũng như với các cơ sở dữ liệu SQL, Apigee, REST APIs,… Điều này cho phép doanh nghiệp kết nối dữ liệu và phần mềm vào một nền tảng thống nhất, tạo điều kiện cho việc triển khai và quản lý ứng dụng một cách hiệu quả.

Với AppSheet, các doanh nghiệp nhỏ cũng có thể cạnh tranh với các doanh nghiệp lớn bằng cách phát triển và triển khai các ứng dụng tùy chỉnh nhanh chóng để đáp ứng nhu cầu cụ thể của họ. Điều này làm cho AppSheet trở thành một công cụ quan trọng trong việc thúc đẩy sự đổi mới và tối ưu hóa quy trình làm việc trong doanh nghiệp.

# **Chương 2: NGHIÊN CỨU VỀ APPSHEET**

## **Giới hạn của Appsheet**

Appsheet là một nền tảng đa năng để tạo ứng dụng di động và web mà không cần phải code. Tuy nhiên, nó cũng có một số giới hạn nhất định giữa phiên bản miễn phí và trả phí mà ta cần lưu ý:

|  |  |  |
| --- | --- | --- |
|  | Phiên bản miễn phí | Phiên bản trả phí |
| Giới hạn dữ liệu | * Giới hạn 10000 hàng cho mỗi bảng dữ liệu. * Giới hạn 20 bảng cho mỗi cơ sở dữ liệu. * Giới hạn 20 cơ sở dữ liệu cho mỗi người dùng. | Loại bỏ các giới hạn của phiên bản miễn phí và cung cấp lượng lưu trữ dữ liệu lớn hơn. |
| Giới hạn tính năng | * Cung cấp các tính năng cơ bản để tạo ứng dụng đơn giản. * Một số tính năng nâng cao bị giới hạn hoặc không khả dụng. | Mở khóa tất cả các tính năng, bao gồm logic nâng cao, tự động hóa, tích hợp API và hơn thế nữa. |
| Giới hạn hiệu suất | Hiệu suất có thể bị ảnh hưởng bởi độ phức tạp của ứng dụng, lưu trữ và cấu trúc kết nối dữ liệu. Tối ưu hóa truy vấn dữ liệu và chia nhỏ các bảng dữ liệu là giải pháp mà Appsheet đưa ra để cải thiện hiệu suất. | |
| Giới hạn tuỳ chỉnh | Khả năng tùy chỉnh giao diện người dùng của ứng dụng bị giới hạn ở một mức độ nào đó. | |
| Hỗ trợ | Phiên bản miễn phí chỉ cung cấp hỗ trợ cộng đồng qua diễn đàn và tài liệu. | Phiên bản trả phí cung cấp hỗ trợ qua email và số điện thoại. |
| Chia sẻ người dùng | Một mail được chia sẻ miễn phí cho 9 mail dùng tất cả các app. | Tăng giới hạn mail có thể chia sẻ |

*Bảng 2.1. Bảng so sánh giới hạn của 2 phiên bản appsheet*

Các gói trả phí:

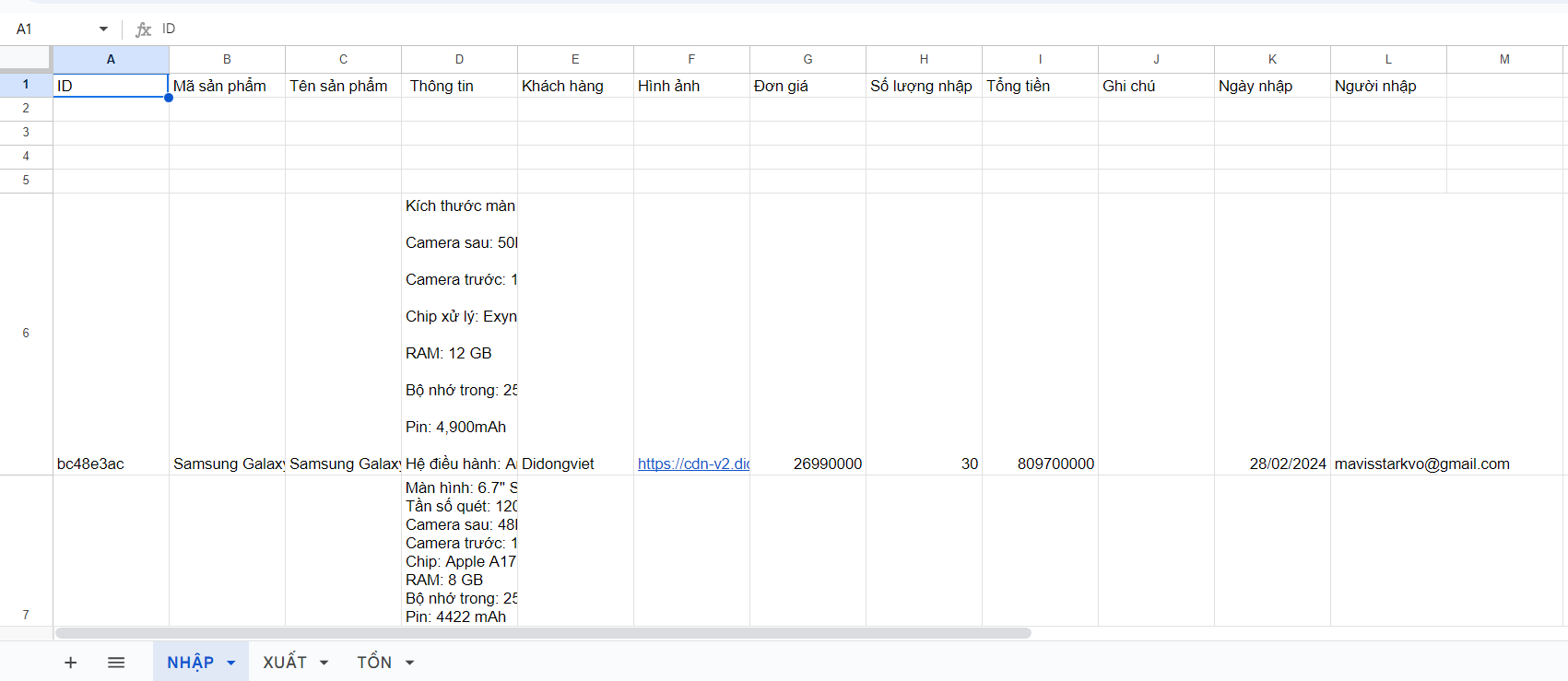
* Stater (Loại thường): 5$/mail/tháng
* Core (Loại trung bình): 10$/mail/tháng
* Public (Loại dành cho app công khai): 50$/app/tháng
* Enterprise (Loại dành cho doanh nghiệp): 25-35$/mail/tháng

Theo như nghiên cứu được các gói nâng cấp càng có giá cao thì sẽ cho người dùng nhiều tính năng hữu ích. Ví dụ như gói Stater có tính năng Quick Sync hay gói Core có tính năng lọc dữ liệu nâng cao cực kì tối ưu cho người dùng.

## **Dữ liệu và cách làm việc với dữ liệu trên Appsheet**

### **Kết nối dữ liệu**

Dữ liệu là nền tảng cho mọi ứng dụng Appsheet. Để có một ứng dụng Appsheet hoàn chỉnh chúng ta cần phải đầu tư khá nhiều cho dữ liệu của mình. Appsheet cũng sẽ cung cấp các công cụ để kết nối với nhiều nền tảng lưu trữ dữ liệu khác nhau: Google Sheets, Airtable, Salesforce, SQL Server, mySQL,….



*Hình 2.1. Dữ liệu được lưu trữ bằng Google Sheets*

Sau khi kết nối dữ liệu, Appsheet sẽ cung cấp các tính năng sau để làm việc với dữ liệu như:

* Tạo các biểu mẫu và bảng: Hiển thị dữ liệu của bạn ở dạng dễ đọc và dễ sử dụng.
* Lọc và sắp xếp dữ liệu: Tìm kiếm thông tin cụ thể một cách nhanh chóng và hiệu quả.
* Thực hiện các phép tính: Tính toán tổng, trung bình, v.v. trên dữ liệu của bạn.
* Tạo biểu đồ và đồ thị: Hình dung dữ liệu của bạn để dễ dàng hiểu hơn.
* Tự động hóa quy trình làm việc: Tự động hóa các tác vụ lặp đi lặp lại để tiết kiệm thời gian và công sức.
* Gửi email và thông báo: Nhận thông báo khi có thay đổi dữ liệu quan trọng.

Trong đề tài này, nhóm em dùng Google Sheets làm bảng dữ liệu vì nó phù hợp với quy mô ứng dụng vừa và nhỏ.

### **Định dạng dữ liệu**

Một cơ sở dữ liệu cơ bản sẽ bao gồm các hàng và cột. Trong đó cái tên thường phản ánh nội dung và bản chất của đối tượng hoặc biến mà nó được dùng để đặt tên. Đặt tên sao cho nó có ý nghĩa đúng với nội dung đã khó mà đặt tên sao cho đúng với quy tắc chuẩn cũng là điều khó không kém. Đặt tên theo quy tắc chuẩn sẽ giúp những người khác khi xem code của bạn cũng cảm thấy dễ hiểu hơn vì nó một quy tắc đã được định sẵn

* Sử dụng một kí tự alphabet ở đầu tên, không sử dụng các kí tự "\_" gạch dưới, hoặc số ở đầu tên biến.
* Giới hạn việc viết tắt quá nhiều trong tên biến hoặc đối tượng. Điều này dễ gây nhầm lẫn.
* Giới hạn việc sử dụng các từ viết tắt hai lần trong tên.
* Đặt tên mang ý nghĩa đúng với nội dung và tính chất của biến hoặc đối tượng.
* Không sử dụng khoảng trắng(space) hoặc các kí tự đặc biệt trong tên như \*,$,#...
* Đặt tên khóa chính nên có hậu tố Id thường thì công thức sẽ là tên bảng + "Id".
* Đối với khóa ngoại thì nên có tên giống với khóa chính mà nó tham chiếu tới. Ví dụ khóa chính của bảng Staff là "StaffId" và bảng [Class] có khóa ngoại đến bảng Staff thì nó có tên là "StaffId".
* Hạn chê sử dụng tiếng Việt có dấu khi đặt mặc dù Appsheet không cấm.

Khi tuân thủ các quy tắc trên ta sẽ dễ dàng thay đổi nền tảng cơ sở dữ liệu khi cần mà không cần tốn công sức để chỉnh sửa tên các cột, hàng.

Trong thực tế, khi thiết kế một ứng dụng, ta không thể tránh được việc người dùng nhập dữ liệu không theo ý mình. Ví dụ ta muốn người dùng nhập 04/05/2024 theo định dạng là dd/MM/YYYY thay vì 4/5/24 hoặc 5/4/2024. Việc nhập liệu không đúng định dạng như này sẽ ảnh hưởng rất nhiều về mặt thời gian cho người lập trình. Sau khi có bảng dữ liệu, việc đầu tiên ta cần làm là phải định dạng các kiểu dữ liệu cho từng cột. Việc này có thể mất đôi chút thời gian những kết quả đạt được lại rất hiệu quả về sau.

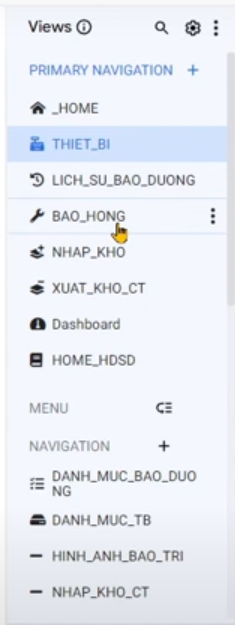
## **View**

Views (Chế độ xem) là một trong những tính năng quan trọng nhất của Appsheet, cho phép bạn hiển thị và tương tác với dữ liệu trong ứng dụng của mình theo nhiều cách khác nhau. Appsheet cung cấp nhiều loại views khác nhau để đáp ứng mọi nhu cầu, bao gồm:

* Dashboard View: Hiển thị tổng quan về dữ liệu của bạn với các biểu đồ, bảng, chỉ số và thẻ.
* Table View: Hiển thị dữ liệu dạng bảng với khả năng sắp xếp, lọc và tìm kiếm.
* Form View: Tạo biểu mẫu để thu thập dữ liệu từ người dùng.
* Detail View: Hiển thị chi tiết cụ thể về một mục dữ liệu.
* Card View: Hiển thị dữ liệu dạng thẻ với hình ảnh, tiêu đề và mô tả.
* Gallery View: Hiển thị hình ảnh, video hoặc các tệp tin khác trong thư viện.
* Map View: Hiển thị dữ liệu trên bản đồ.
* Deck View: Hiển thị dữ liệu dạng thẻ có thể lật sang để xem thêm thông tin.
* Onboarding View: Giới thiệu ứng dụng và hướng dẫn người dùng sử dụng.

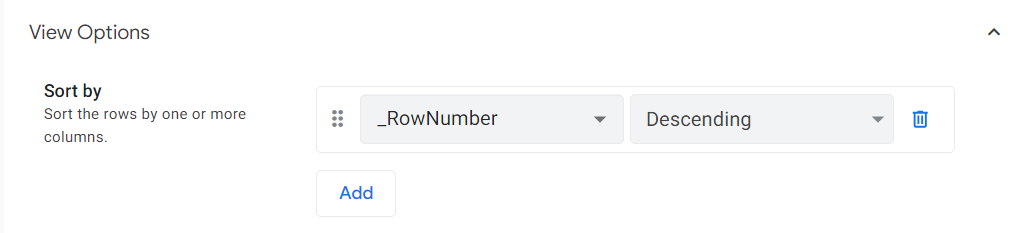
Khi thao tác trên views ta cần lưu ý những điều như sau:

* Trên ứng dụng, tên view nên được đặt tên theo tên bảng (table) mà view đó sẽ hiển thị. Ví dụ: nếu ta tạo view cho table SANPHAM, thì tên table sẽ là SANPHAM hoặc SANPHAMUSER. Phần display name có thể được đặt theo tuỳ thích.



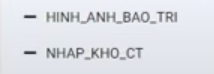
*Hình 2.2. Tên views*

Ở đây chúng ta chọn phần views trên thanh công cụ để xem tất cả những view hiện có. Trong phần hiển thị này cho thấy những phần view nào đang hiển thị cho table nào. Điều đó giúp cho quá trình làm việc trở nên dễ dàng hơn rất nhiều.



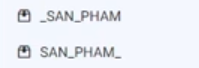
*Hình 2.3. View options*

* Tiếp đến trong phần view options ta nên thiết lập sort by là \_Rownumber và chọn decending để hiển thị các dòng mới nhất lên đầu tiên.



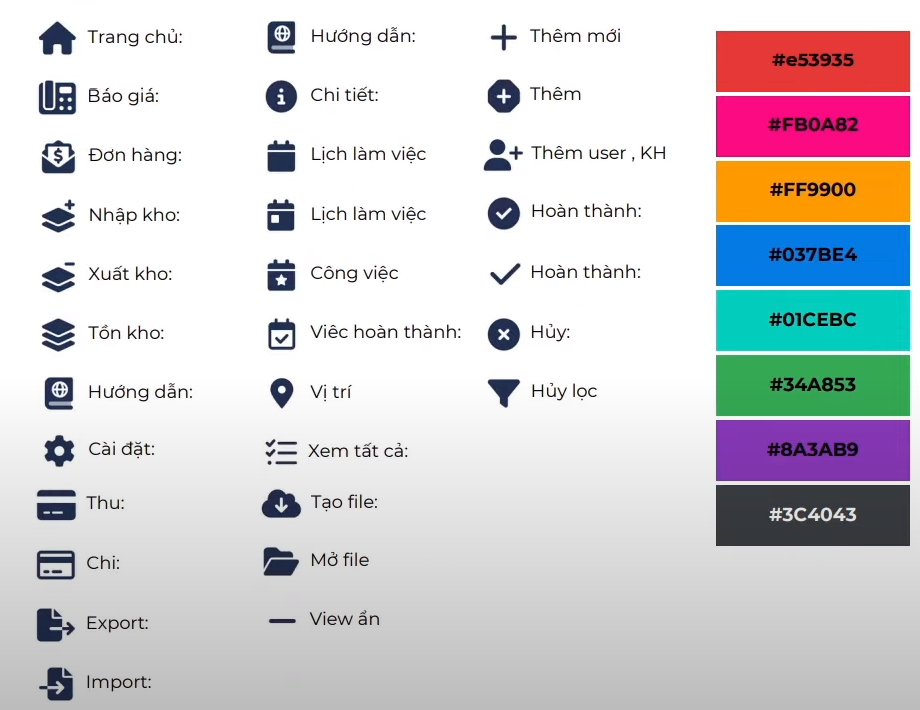
*Hình 2.4. Icon minus đối với các views không cần cho người nhìn thấy*

* Icon cho views ẩn nên được hiển thị với icon minus



*Hình 2.5. Icon hiển thị đồng nhất*

* Đồng nhất icon: các icon nên có hình ảnh đồng nhất với nhau. Ví dụ: vuông, tròn hoặc hình ảnh tương tự nhau như hình 2.5.



*Hình 2.6. Sử dụng màu trong view*

* Cuối cùng là màu sắc trên app: Nên sử dụng tối đa 5 màu trên một view để người dùng dễ nhìn và không bị nhầm lẫn.

## **Actions**

Actions là một tính năng mạnh mẽ trong Appsheet cho phép bạn tự động hóa các tác vụ, mở rộng chức năng và tạo ra các ứng dụng phức tạp hơn. Actions có thể được sử dụng để thực hiện nhiều loại nhiệm vụ, bao gồm:

* Điều hướng giữa các views: Di chuyển người dùng đến các màn hình khác trong ứng dụng của bạn.
* Thay đổi dữ liệu: Thêm, xóa, cập nhật dữ liệu trong các bảng.
* Gửi email: Gửi email cho người dùng hoặc các bên liên quan khác.
* Tạo báo cáo: Tạo báo cáo PDF hoặc CSV từ dữ liệu của bạn.
* Gọi API: Kết nối với các API bên ngoài để truy cập dữ liệu và dịch vụ khác.
* Tự động hóa quy trình làm việc: Thực hiện các tác vụ lặp đi lặp lại một cách tự động.

Có bốn loại Actions chính trong Appsheet:

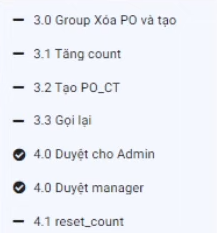
* User Actions: Được kích hoạt bởi người dùng, chẳng hạn như khi họ nhấp vào nút hoặc nhập dữ liệu vào biểu mẫu.
* Data Actions: Được kích hoạt bởi sự thay đổi dữ liệu, chẳng hạn như khi thêm hoặc cập nhật dữ liệu trong bảng.
* Schedule Actions: Được thực hiện tự động theo lịch trình, chẳng hạn như vào một thời điểm cụ thể mỗi ngày hoặc mỗi tuần.
* External Actions: Gọi API bên ngoài để truy cập dữ liệu và dịch vụ khác.

Tạo Actions:

* Chọn loại Action bạn muốn tạo.
* Cung cấp thông tin cần thiết cho Action, chẳng hạn như dữ liệu cần thay đổi, email cần gửi hoặc API cần gọi.
* Thêm logic nếu cần thiết, chẳng hạn như điều kiện hoặc vòng lặp.
* Lưu Actions và thử nghiệm.

Sau khi chúng ta đã tiếp hiểu các lưu ý liên quan đến dữ liệu và hiển thị, bây giờ ta sẽ đến phần lưu ý của action.

* Đối với action bị ẩn: chúng ta sẽ sử dụng biểu tượng minus như hình 2.4



*Hình 2.7. Đánh số thứ tự cho các action*

* Nhóm action: Khi có nhiều hành động được nhóm lại, chúng cần được đánh số theo thứ tự như hình 2.7.
* Đồng bộ hoá app: Các action cần được sắp xếp theo thứ tự cụ thể. Ví dụ: action tạo file cần được đặt trước action mở file. Thứ tự này cần được tuân thủ trong tất cả các bảng của ứng dụng.
* Đối với màu sắc: Nó cũng cần được tuân thủ cho tất cả các hành động. Ví dụ: action duyệt sẽ có màu xanh còn huỷ sẽ có màu đỏ.
* Tên action nên được đặt đồng nhất.

## **Automation**

Automation là một tính năng trong Appsheet cho phép bạn tự động hóa các tác vụ lặp đi lặp lại, tối ưu hóa quy trình làm việc và tiết kiệm thời gian.

Lợi ích của Automation:

* Tiết kiệm thời gian: Tự động hóa các tác vụ lặp đi lặp lại giúp bạn tiết kiệm thời gian và công sức, cho phép bạn tập trung vào những công việc quan trọng hơn.
* Nâng cao hiệu quả: Automation giúp tối ưu hóa quy trình làm việc, loại bỏ sai sót do con người và tăng hiệu quả hoạt động.
* Giảm thiểu lỗi: Automation giúp giảm thiểu sai sót do con người, đảm bảo tính chính xác và nhất quán trong dữ liệu.
* Mở rộng khả năng: Automation giúp bạn tạo ra các ứng dụng phức tạp hơn và mở rộng chức năng của ứng dụng.

Cách sử dụng Automation:

* Xác định tác vụ cần tự động hóa: Xác định các tác vụ lặp đi lặp lại mà bạn muốn tự động hóa.
* Chọn loại Automation: Chọn loại Automation phù hợp với nhu cầu của bạn, chẳng hạn như User Automation, Data Automation, Schedule Automation hoặc External Automation.
* Cung cấp thông tin cần thiết: Cung cấp thông tin cần thiết cho Automation, chẳng hạn như dữ liệu cần thay đổi, email cần gửi hoặc API cần gọi.
* Thêm logic: Thêm logic nếu cần thiết, chẳng hạn như điều kiện hoặc vòng lặp.
* Lưu Automation và thử nghiệm: Lưu Automation và thử nghiệm để đảm bảo nó hoạt động chính xác.

Ví dụ:

* Tự động tạo đơn hàng: Khi khách hàng gửi biểu mẫu đơn hàng, tự động tạo đơn hàng mới trong hệ thống quản lý bán hàng.
* Gửi email thông báo: Khi có đơn hàng mới, tự động gửi email thông báo cho bộ phận kho để chuẩn bị hàng hóa.
* Cập nhật dữ liệu kho: Khi đơn hàng được giao, tự động cập nhật số lượng tồn kho trong hệ thống quản lý kho.
* Gửi báo cáo doanh số: Gửi báo cáo doanh số bán hàng theo ngày, tuần hoặc tháng cho quản lý.

## **Các chức năng nâng cao trong Appsheet**

AppSheet cung cấp nhiều tính năng miễn phí mạnh mẽ để bạn bắt đầu tạo ứng dụng di động và web. Tuy nhiên, để mở rộng khả năng và tạo ra các ứng dụng phức tạp hơn, ta cần nâng cấp lên gói trả phí:

### **Tính năng nâng cao về dữ liệu:**

* Kết nối với nhiều nguồn dữ liệu: Kết nối với nhiều nguồn dữ liệu hơn, bao gồm cơ sở dữ liệu SQL, API RESTful và các tệp tin đám mây.
* Truy vấn dữ liệu phức tạp: Thực hiện truy vấn dữ liệu phức tạp hơn với các bộ lọc, sắp xếp và kết hợp nâng cao.
* Tạo báo cáo và biểu đồ tùy chỉnh: Tạo báo cáo và biểu đồ tùy chỉnh từ dữ liệu của bạn để trực quan hóa thông tin.
* Machine Learning: Tích hợp các mô hình học máy vào ứng dụng của bạn để phân tích dữ liệu và đưa ra dự đoán.

### **Tính năng nâng cao về ứng dụng:**

* Actions nâng cao: Tạo Actions phức tạp hơn với logic tùy chỉnh, kết nối API và xử lý dữ liệu.
* Logic nâng cao: Kiểm soát hành vi ứng dụng bằng cách sử dụng các biểu thức logic phức tạp hơn, vòng lặp và xử lý lỗi.
* Custom Code: Mở rộng khả năng của AppSheet bằng cách thêm mã JavaScript hoặc TypeScript tùy chỉnh để tạo các chức năng và hành vi ứng dụng độc đáo.
* Progressive Web Apps (PWAs): Tạo các ứng dụng web có thể cài đặt và hoạt động như ứng dụng di động với các tính năng nâng cao như truy cập ngoại tuyến và thông báo đẩy.
* Bảo mật nâng cao: Bảo vệ dữ liệu ứng dụng của bạn với các tính năng bảo mật nâng cao như xác thực đa yếu tố, mã hóa dữ liệu tại chỗ và kiểm soát truy cập dựa trên vai trò.

### **Tính năng quản lý và cộng tác:**

* Quản lý người dùng: Quản lý người dùng và quyền truy cập ứng dụng với các tính năng như xác thực người dùng, phân quyền vai trò và nhật ký truy cập.
* Cộng tác nhóm: Cộng tác với các thành viên trong nhóm để phát triển và quản lý ứng dụng với các tính năng như chia sẻ ứng dụng, chỉnh sửa đồng thời và quản lý phiên bản.
* Hỗ trợ nâng cao: Nhận hỗ trợ ưu tiên từ nhóm hỗ trợ AppSheet để giải quyết các vấn đề và nhận trợ giúp với các tính năng nâng cao.

# **Chương 3: ỨNG DỤNG APPSHEET ĐỂ XÂY DỰNG ỨNG DỤNG QUẢN LÝ KHO HÀNG**

## **Đặc tả**



*Hình 3.1. Đặc tả ứng dụng*

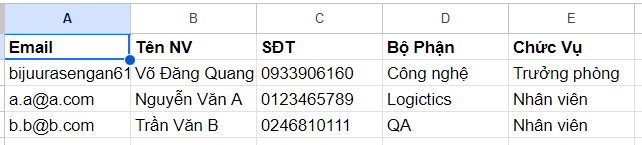
Ta sẽ bắt đầu với quản lý thông tin, đây là các bảng dữ liệu được dùng vào ứng dụng. Trong đó 3 bảng dữ liệu đó là: người dùng, danh sách kho và sản phẩm sẽ là 3 bảng dữ liệu chính dùng để thiết kế ứng dụng.

### **Mô tả ứng dụng**

* Mục tiêu dự án: Xây dựng ứng dụng quản lý kho hàng.
* Chi tiết mục tiêu: Tạo một ứng dụng có các chức năng như viết phiếu nhập, xuất; kiểm tra hàng tồn; báo cáo nhập – xuất – tồn; cảnh báo tồn kho;….
* Đối tượng sử dụng: Người quản lý kho.
* Thời gian hoàn thành: 3 tháng.
* Giao diện: Dễ dùng, đơn giản, thân thiện.
* Mức độ bảo mật: Không yêu cầu cao.
* Hệ điều hành: Windows, MacOS, Linux.
* Phần cứng: Không yêu cầu cao.
* Công nghệ: Appsheet, Telegram.
* Nhóm nghiệp vụ:
* Quản lý nhập xuất
* Điều chỉnh, kiểm kê
* Báo cáo nhập – xuất – tồn
* Yêu cầu khác: App có khả năng bảo trì và nâng cấp dễ dàng.

#### **Quản lý người dùng**

Đối với người dùng, ta phải có các thông tin như tên, email, số điện thoại,… để lưu trữ thông tin.



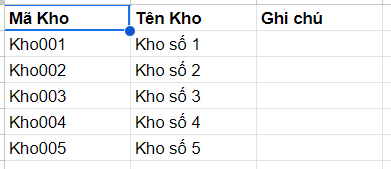
*Hình 3.2. Bảng dữ liệu người dùng*

Việc lưu trữ các thông tin như “Bộ phận” hay “Chức vụ” sẽ giúp cho các nhà phát triển dễ dàng hơn trong việc phần quyền người dùng nếu như về sau khách hàng có yêu cầu.

#### **Quản lý danh sách kho**

Đối với danh sách kho, ta sẽ có các thông tin bắt buộc là:

* “Mã kho” để làm khoá cho bảng dữ liệu này.
* “Tên kho” là thông tin về tên nhà kho ứng với “Mã kho”.
* “Ghi chú” là dữ liệu để lưu lại các ghi chú mà thủ kho ghi lại như giới hạn tồn hay các lưu ý khi lưu kho.



*Hình 3.3. Bảng dữ liệu danh sách kho*

#### **Quản lý sản phẩm**

Cuối cùng sản phẩm là bảng dữ liệu quan trọng nhất của ứng dụng này.



*Hình 3.4. Bảng dữ liệu sản phẩm*

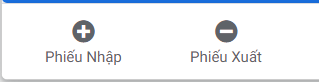
Trong bảng dữ liệu này ta phải chứa các thông tin quan trọng như:

* “Mã Sản Phẩm” để làm khoá cho bảng dữ liệu này.
* “Tên Sản Phẩm” để lưu tên sản phẩm.
* “Mô tả” để lưu các mô tả đặc điểm của sản phẩm.
* “Đơn vị tính”
* “Giới hạn trên” và “ Giới hạn dưới” là 2 dữ liệu thể hiện giới hạn tồn cho mỗi sản phẩm.

### **Các chức năng ứng dụng**

#### **Quản lý nhập xuất**

Đây là chức năng chính đầu tiên của ứng dụng. Vì đây là ứng dụng quản lý kho thì đương nhiên nó phải diễn ra các hoạt động như nhập, xuất, luân chuyển hàng hoá trong kho,…



*Hình 3.5. Chức năng nhập xuất của ứng dụng*

Ta sẽ phải tạo ra các chức năng để ghi lại những điều đó. Ngoài ra ta còn phải tạo các chức năng xuất ra phiếu nhập/xuất kho để cho người lập, người nhận ký vào rồi lưu trữ các tài liệu này theo một quy trình nào đó.



*Hình 3.6. Phiếu nhập kho*

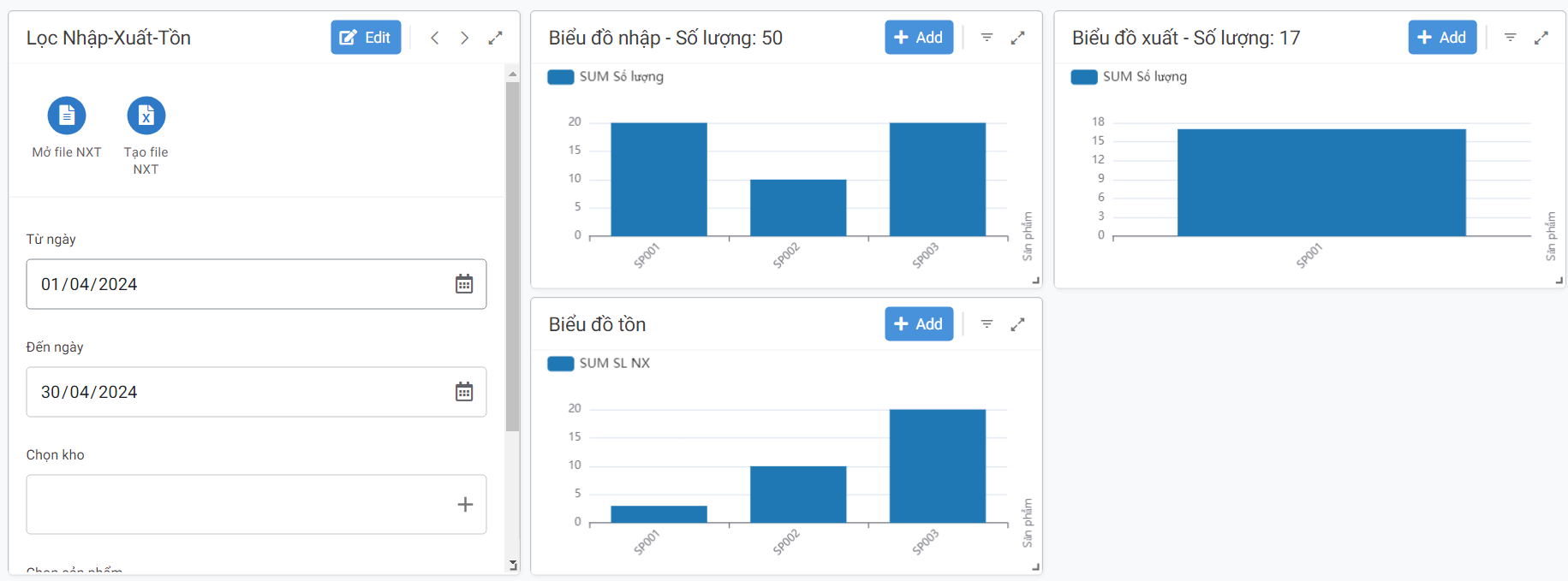
#### **Điều chỉnh, kiểm kê**

Chức năng chính tiếp theo đó là điều chỉnh và kiểm kê. Trong quản lý kho hàng thì không thể tránh khỏi việc sai lệch số liệu giữa sổ sách với thực tế. Chính vì vậy nên ta cần phải tạo ra chức năng cho phép kiểm kê định kì hằng tháng có khi là hằng tuần để điều chỉnh dữ liệu, đảm bảo giảm tối thiểu tỉ lệ xảy ra sai sót đó.

Một chức năng quan trọng nữa đó là backup dữ liệu. Khoảng 1 đến 2 năm ta nên backup dữ liệu một lần. Nếu để quá lâu không backup dữ liệu thường sẽ xảy ra trường hợp load ứng dụng rất chậm vì mỗi lần truy cập sẽ phải load dữ liệu của rất nhiều năm trước, ảnh hưởng đến việc sử dụng.

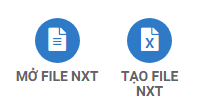
#### **Báo cáo nhập – xuất – tồn**

Tiếp theo là chức năng báo cáo nhập – xuất – tồn, ta phải báo cáo theo một hình thức nào đó cho nó đầy đủ, trực quan, nhanh và phân tích được sâu như theo từng giai đoạn, theo kho hay theo từng sản phẩm.

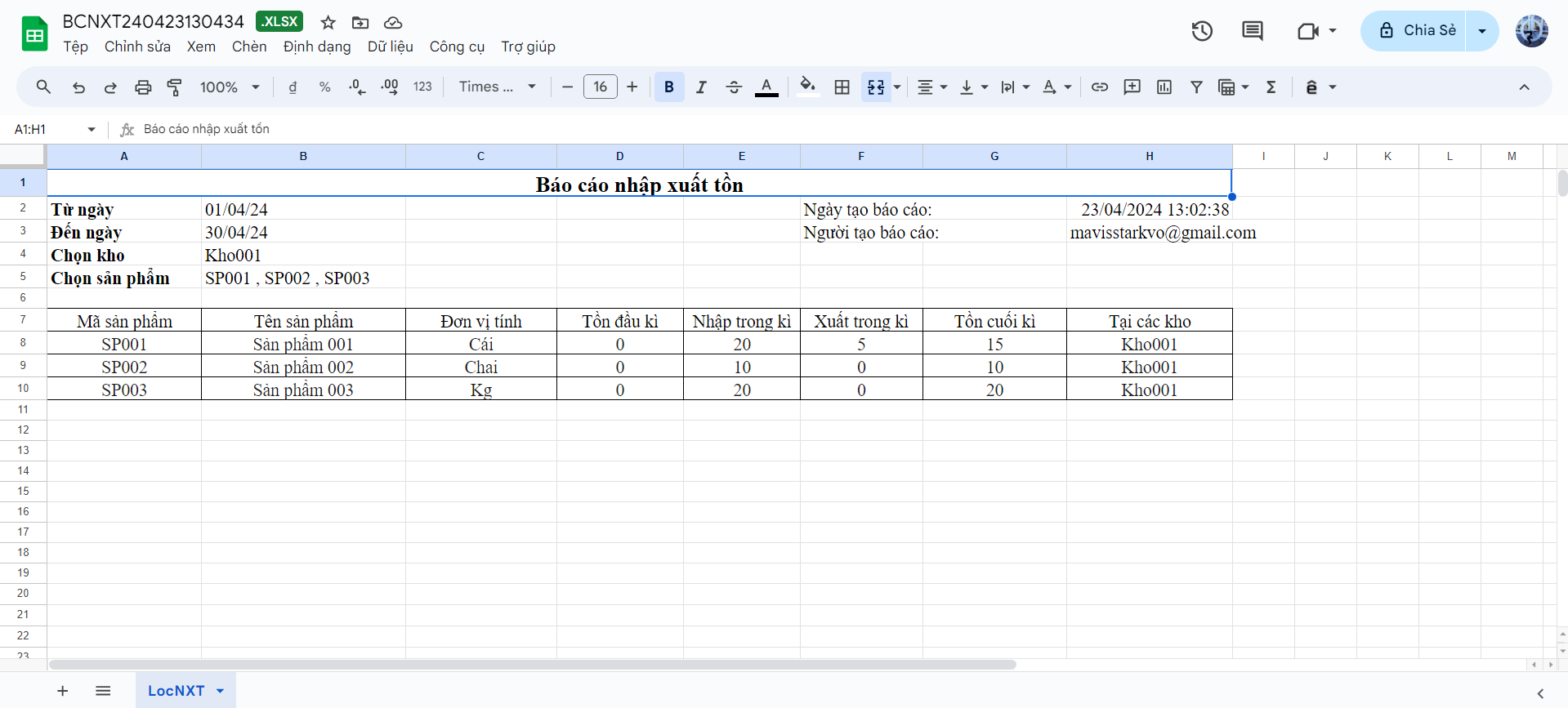


*Hình 3.7. Trực quan hoá báo cáo nhập - xuất - tồn*

Bên cạnh đó ta nên có các chức năng như cảnh báo tồn hay in báo cáo nhập – xuất – tồn ra file Excel.



*Hình 3.8. Chức năng xuất ra file Excel*



*Hình 3.9. File báo cáo Excel*

## **Tạo cơ sở dữ liệu**

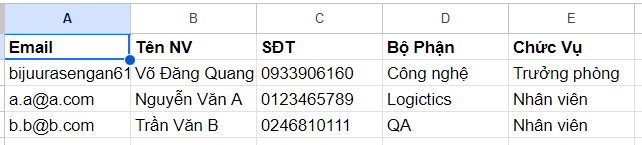
Để tạo cơ sở dữ liệu, ta vào Google Sheets và tạo 1 trang tính mới. Trong đó ta tạo các bảng dữ liệu với tên như hình dưới.



*Hình 3.10. Các bảng dữ liệu trong ứng dụng*

Trong mỗi bảng dữ liệu gồm có rất nhiều trường nhưng quan trọng nhất phải có cột khoá chính. Khoá chính là cột định danh cho mỗi bảng dữ liệu, ta cần đảm bảo rằng nó là duy nhất và không có một dữ liệu nào trùng với khoá chính.

### **Bảng dữ liệu User**



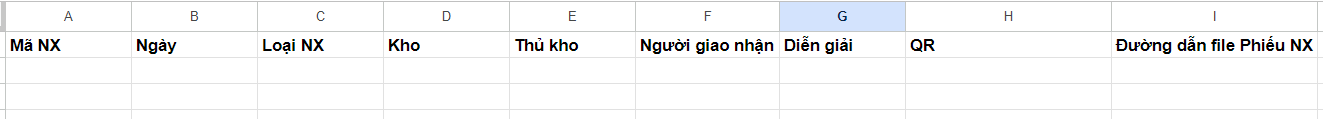
*Hình 3.11. Bảng dữ liệu User*

Trong bảng dữ liệu User này, khoá chính được chọn mà “Email”. Mỗi người có một email nhất định và không ai giống ai. Chính vì vậy chọn email làm khoá chính là hợp lý nhất. Sau khi có khoá chính ta thêm trường dữ liệu khác để lưu thông tin người dùng như:

* Tên người dùng là “Tên NV”.
* Số điện thoại người dùng: “SĐT”.
* Người dùng thuộc bộ phận nào: “Bộ Phận”.
* Chức vụ của người dùng để dễ phân quyền hơn sau này: “Chức Vụ”.

Trong ứng dụng này nhóm sử dụng 5 trường dữ liệu là đủ để quản lý thông tin người dùng. Ngoài ra ta còn có thể thêm một số thông tin như căn cước công dân, ngày sinh, mã số thuế hay ngày vào làm.

### **Bảng dữ liệu Phieu NX**

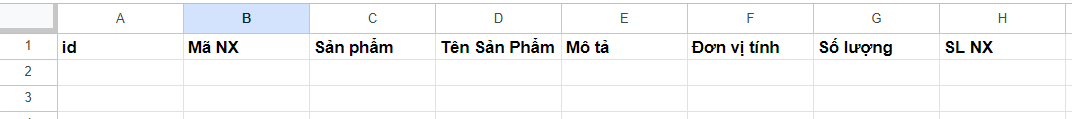


*Hình 3.12. Bảng dữ liệu Phieu NX*

Phieu NX là bảng chứa các thông tin về các thao tác xuất nhập trong ứng dụng. Mỗi một phiếu xuất kho hoặc phiếu nhập kho đều có một mã riêng biệt, đó là khoá chính. Các thuộc tính có trong bảng dữ liệu này:

* “Mã NX” được chọn làm khoá chính
* “Ngày” là ngày lập phiếu nhập/xuất
* “Loại NX”: có 2 kiểu là nhập hoặc xuất
* “Kho”: tên kho
* “Thủ kho” ở đây sẽ lấy tên người dùng làm tên người thủ kho
* “Người giao nhận” là tên người giao nhận. Ở đây ta tự nhập.
* “Diễn giải” là kiểu xuất nhập kho. Ví dụ: Nhập kho có thể là nhập chuyển kho hay nhập điều chỉnh; Xuất kho có thể là xuất sản xuất hay xuất điều chỉnh
* “QR”: trường này sẽ là nơi lưu trữ mã QR code
* “Đường dẫn file Phiếu NX”: trường này sẽ là nơi lưu trữ đường dẫn chứa phiếu nhập xuất có đuôi là .PDF

### **Bảng dữ liệu PhieuNXChitiet**

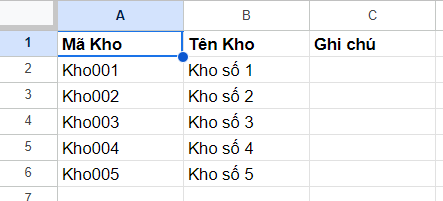


*Hình 3.13. Bảng dữ liệu PhieuNXChitiet*

PhieuNXChitiet là bảng dữ liệu chứa các ghi chép chi tiết hơn của các thao tác nhập xuất của ứng dụng. Các trường trong bảng phiếu nhập xuất chi tiết này thể hiện cho thông tin một phiếu nhập hoặc xuất như: mã sản phẩm, tên sản phẩm, đơn vị tính, số lượng. Các thuộc tính có trong bảng dữ liệu này như sau:

* “id” là khoá chính của bảng dùng để phân biệt các phiếu với nhau.
* “Mã NX” là khoá ngoại dùng để phân biệt phiếu nhập và phiếu xuất.
* “Sản phẩm” là khoá ngoại dùng để thể hiện đó là mã sản phẩm nào.
* “Tên Sản Phẩm” là thuộc tính thể hiện tên sản phẩm.
* “Mô tả” là thuộc tính dùng để mô tả sản phẩm.
* “Đơn vị tính” là thuộc tính mô tả đơn vị tính của sản phẩm.
* “Số lượng” là thuộc tính mô tả số lượng của sản phẩm.
* “SL NX” là thuộc tính mô tả số lượng nhập xuất của sản phẩm.

### **Bảng dữ liệu DSKho**

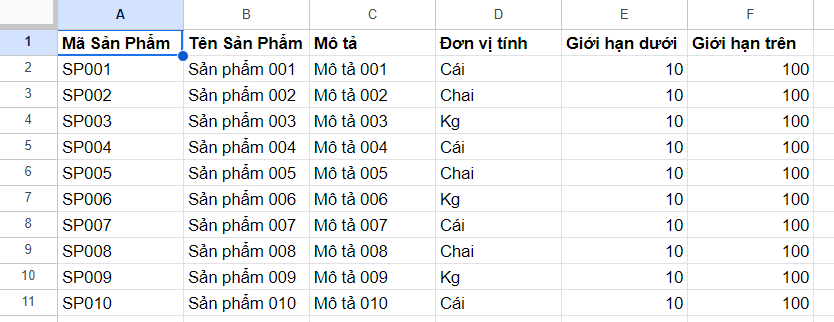


*Hình 3.14. Bảng dữ liệu DSKho*

DSKho là bảng dữ liệu chứa các thông tin liên quan đến kho hàng. Trong bảng dữ liệu này, ta có các trường như sau:

* “Mã Kho” là khoá chính dùng để phân biệt các kho với nhau.
* “Tên Kho” là thuộc tính mô tả tên kho.
* “Ghi chú” là thuộc tính của các dòng ghi chú của từng kho.

### **Bảng dữ liệu SanPham**



*Hình 3.15. Bảng dữ liệu SanPham*

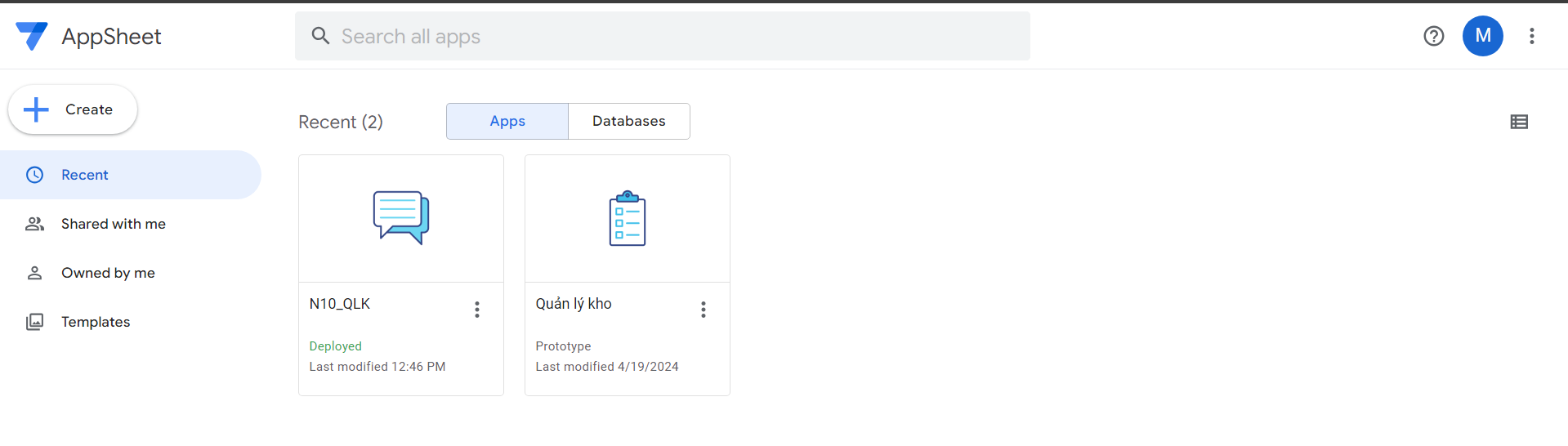
SanPham là bảng chứa các thông tin về các sản phẩm mà ta phải quản lý. Các thuộc tính có trong bảng này diễn tả thông tin sản phẩm như: tên sản phẩm, mô tả sản phẩm, đơn vị tính. Ta có những thuộc tính như sau:

* “Mã Sản Phẩm” là khoá chính của bảng dùng để phân biệt các sản phẩm với nhau.
* “Tên Sản Phẩm” là thuộc tính miêu tả tên của sản phẩm.
* “Mô tả” là thuộc tính mô tả sản phẩm.
* “Đơn vị tính” là thuộc tính mô tả đơn vị tính của sản phẩm.
* “Giới hạn dưới” và “Giới hạn trên” là hai thuộc tính thể hiện giới hạn tồn của sản phẩm

## **Tạo ứng dụng**

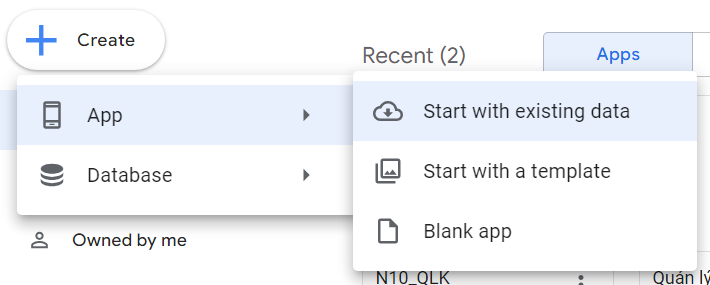
Sau khi đã có các bảng dữ liệu tiếp đến ta sẽ tiến hành tạo giao diện cho ứng dụng. Các bước thực hiện rất đơn giản theo quy trình như sau:

* Bước 1: Truy cập vào trang web <https://www.appsheet.com/>
* Bước 2: Dùng mail của mình đăng nhập vào Appsheet để có thể sử dụng nền tảng này. Đăng nhập xong ta sẽ có giao diện như hình dưới



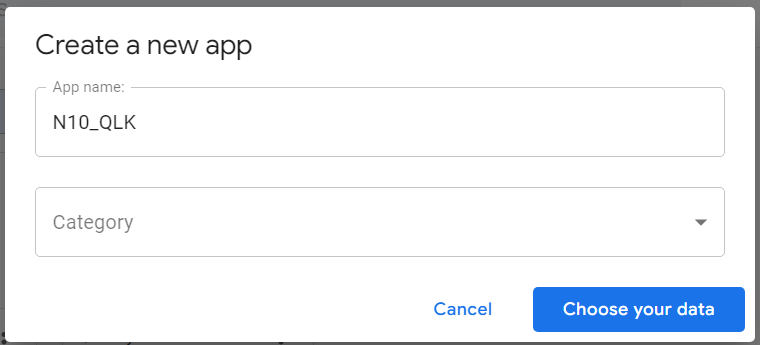
*Hình 3.16. Trang chủ Appsheet*

* Bước 3: Chọn Create > App > Start with existing data để tạo 1 App mới



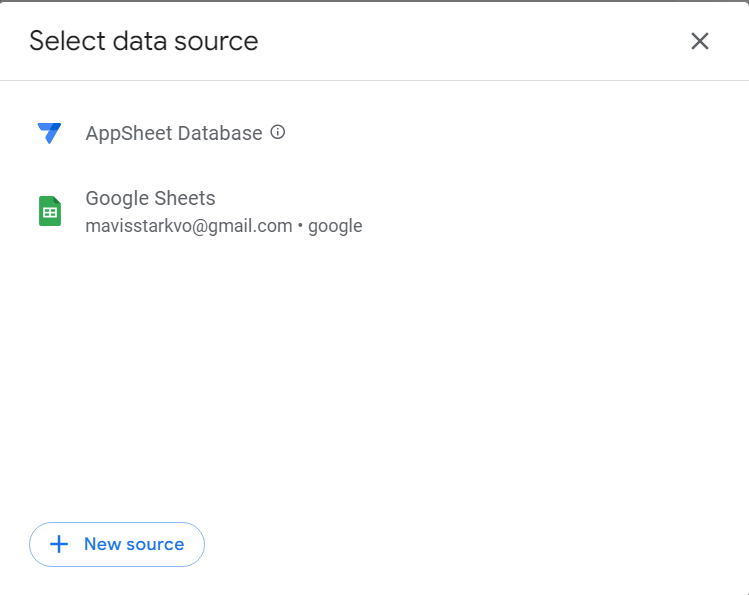
*Hình 3.17. Tạo 1 App mới*

* Bước 4: Ở bước này ta chỉ cần đặt tên cho App và chọn Choose your data để qua bước tiếp theo. Ở đây nhóm sẽ chọn tên App giống với tên của cơ sở dữ liệu là “N10\_QLK”.



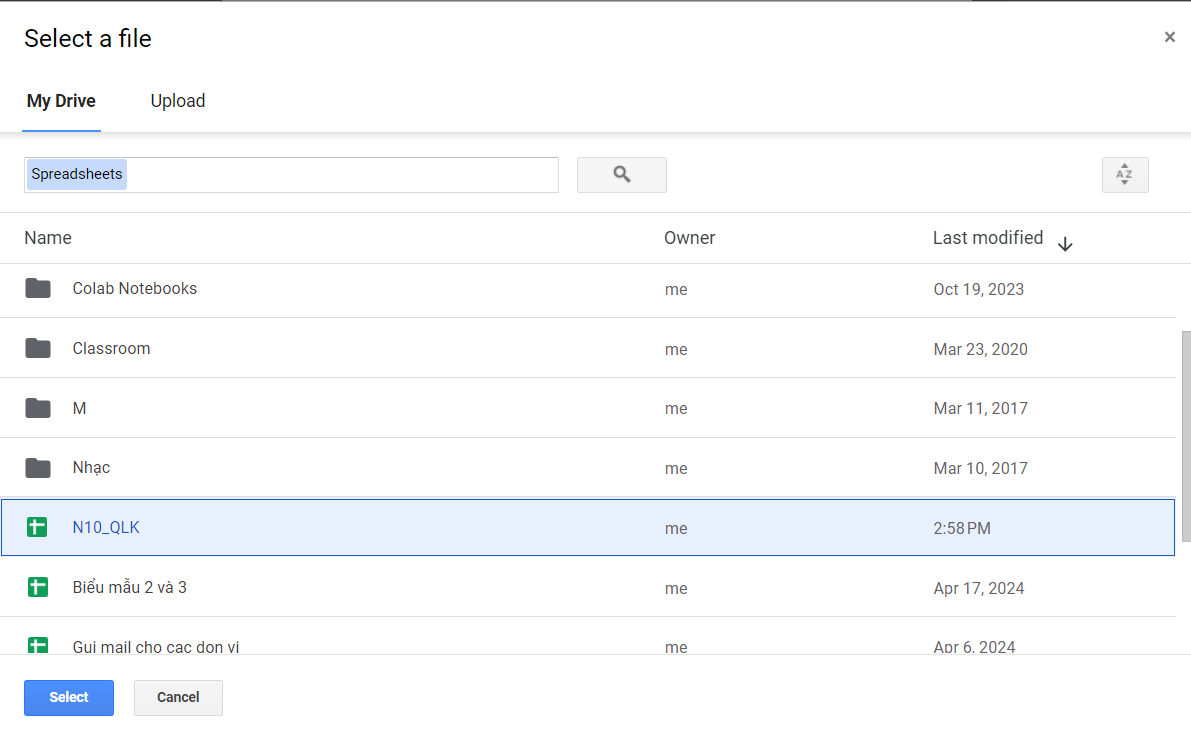
*Hình 3.18. Đặt tên cho App*

* Bước 5: Vì sử dụng Google Sheets làm cơ sở dữ liệu nên ta sẽ chọn Google Sheets.



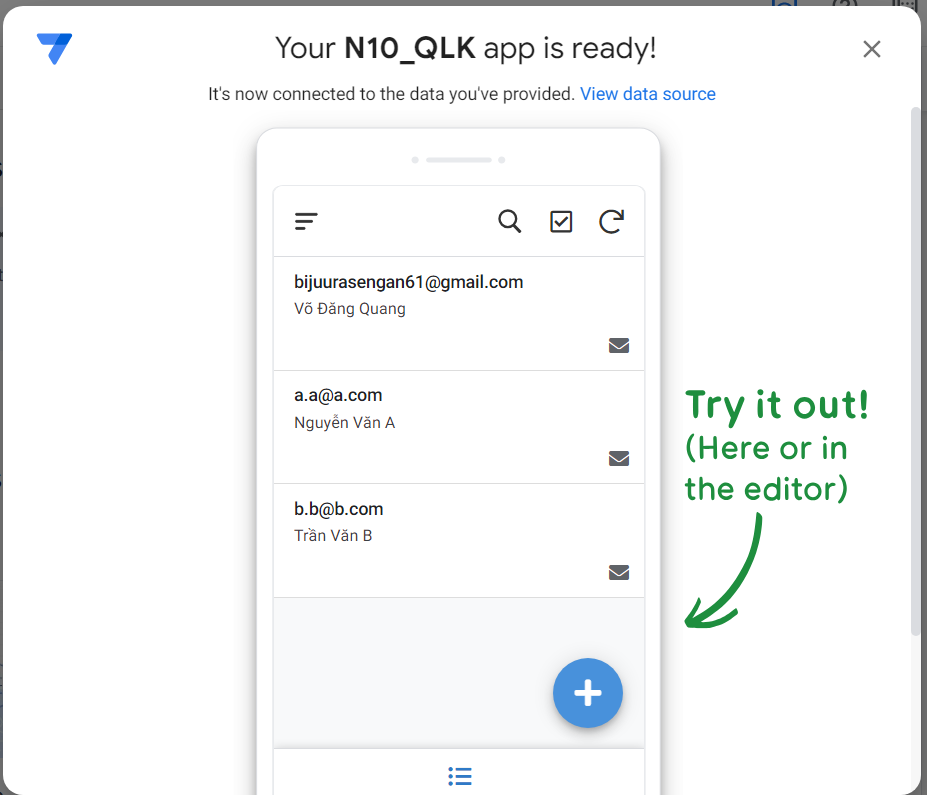
*Hình 3.19. Chọn loại cơ sở dữ liệu*

* Bước 6: Đây là bước cuối cùng. Trong bước này ta phải tìm cơ sở dữ liệu và chọn



*Hình 3.20. Chọn cơ sở dữ liệu đã tạo sẵn*

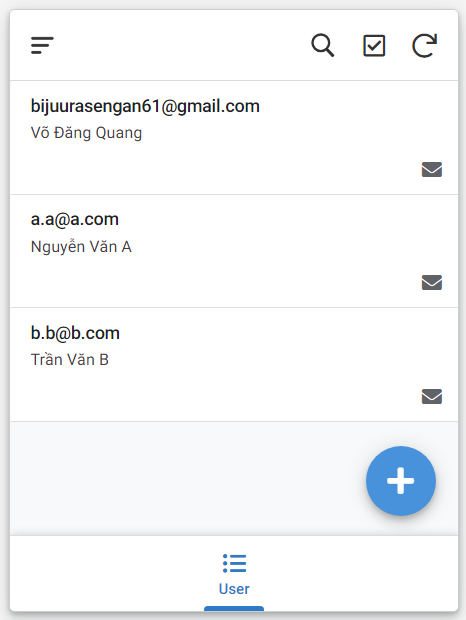
Sau một khoảng thời gian chờ đợi thì Appsheet sẽ tự động tạo cho chúng ta App hoàn chỉnh. Công việc lúc này của các nhà phát triển đó là đi điều chỉnh App sao cho phù hợp với yêu cầu của khách hàng.



*Hình 3.21. App được tạo thành công*

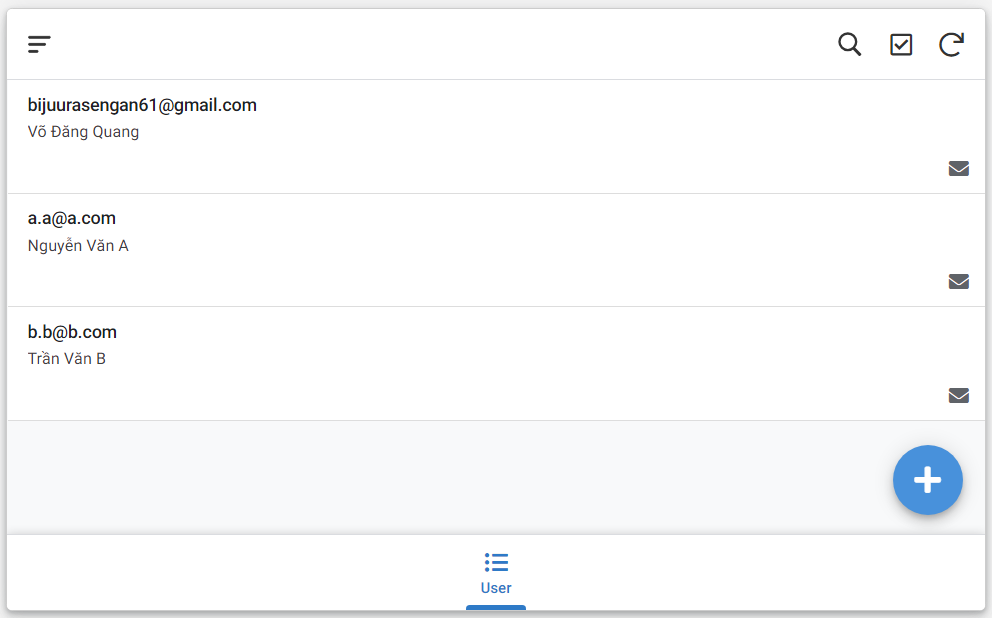
App được tạo bởi Appsheet sẽ được tối ưu theo 3 kiểu để phù hợp với đa số các thiết bị phổ biến hiện nay:

* Giao diện vertical mobile:



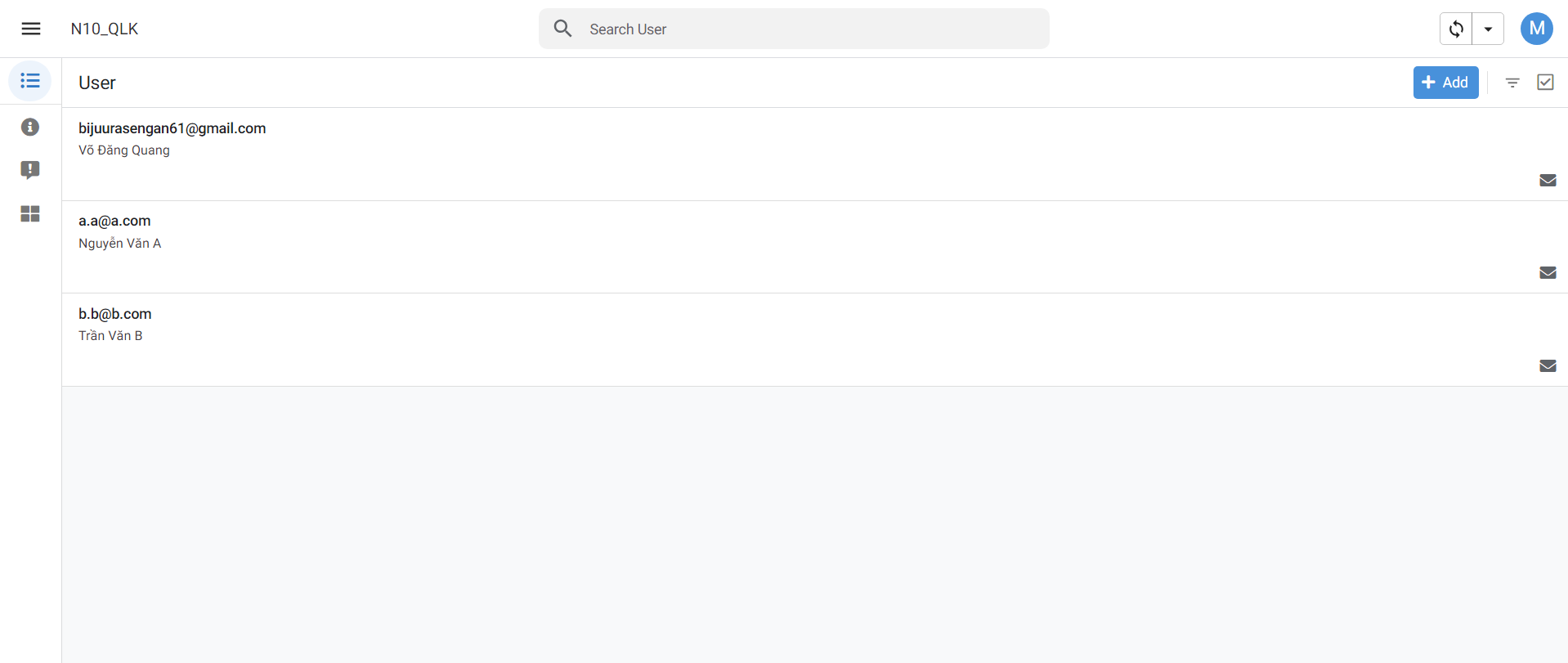
*Hình 3.22. Giao diện vertical mobile*

* Giao diện horizontal mobile:



*Hình 3.23. Giao diện horizontal mobile*

* Giao diện desktop:



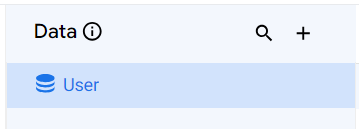
*Hình 3.24. Giao diện desktop*

## **Giao diện quản lý thông tin**

### **Các thiết lập và ràng buộc dữ liệu**

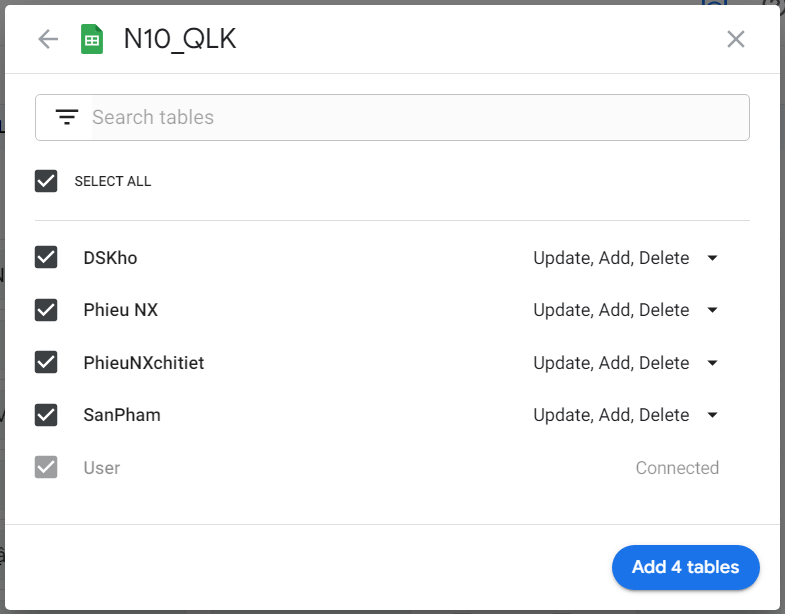
Sau khi ta có giao diện cơ bản mà Appsheet tạo cho, lúc này ứng dụng vẫn chưa thể đi vào hoạt động được. Bước đầu tiên đó là phải có các kiểu dữ liệu và ràng buộc cho từng kiểu dữ liệu trong bảng trước khi sử dụng. Bước này là bước quan trọng nhất quyết định ứng dụng có hoạt động một cách ổn định hay không.

Trên giao diện Appsheet lúc này chỉ mới hiện mỗi bảng dữ liệu User. Ta cần phải thêm các bảng dữ liệu còn lại.



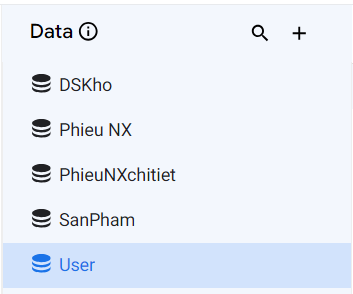
*Hình 3.25. Cơ sở dữ liệu chỉ có mỗi bảng User*

Để có thể thêm các bảng dữ liệu còn lại ta chọn dấu “**+**”, làm lại bước 5 và 6 của phần “**Tạo ứng dụng”** ở trên.



*Hình 3.26. Thêm cấc bảng dữ liệu khác vào cơ sở dữ liệu*

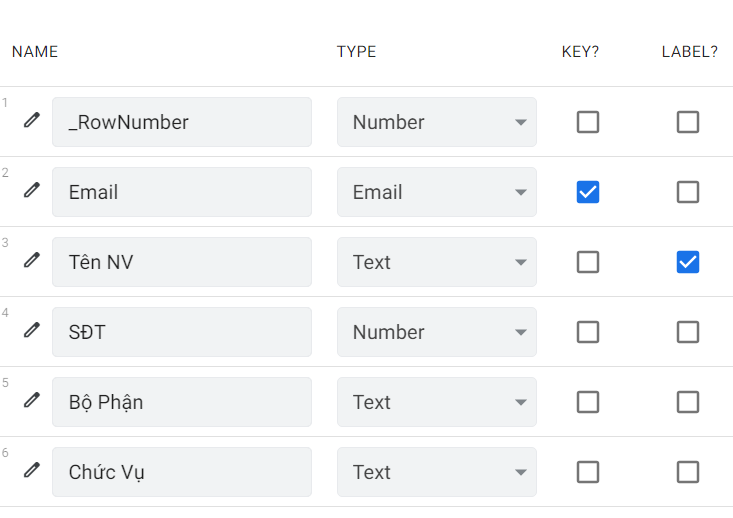
Bước tiếp theo ta chọn các bảng dữ liệu cần thêm vào ứng dụng. Ở đây Appsheet sẽ chọn trước tất cả các bảng cho chúng ta. Các bảng được thêm vào sẽ có quyền cập nhật (**Update**), Thêm (**Add**) và Xoá (**Delete**) dữ liệu. Sau khi hoàn thành các bước trên, cơ sở dữ liệu sẽ như sau:



*Hình 3.27. Cơ sở dữ liệu đã được thêm đầy đủ*

Bây giờ ta đã có một cơ sở dữ liệu đầy đủ, trước khi tạo giao diện cho ứng dụng ta phải qua một bước nữa là điều chỉnh các kiểu dữ liệu cho từng bảng.

#### **Điều chỉnh bảng User**

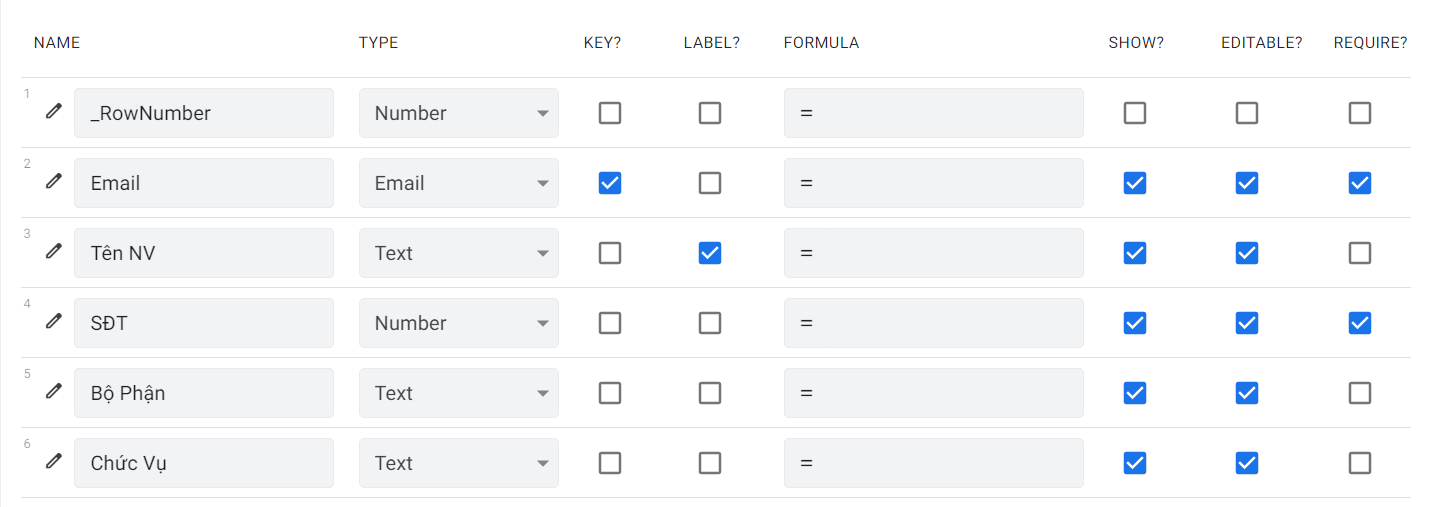


*Hình 3.28. Các trường dữ liệu trong bảng User*

Trong bảng User ta sẽ chỉnh các kiểu dữ liệu như hình 3.28, cụ thể như sau:

* **“Email**“ sẽ có kiểu dữ liệu là Email và ta sẽ set “**Email**” sẽ là khoá chính.
* “**Tên NV**” có kiểu dữ liệu là Text.
* “**SĐT**” có kiểu dữ liệu là Number.
* “**Bộ Phận**” có kiểu dữ liệu là Text.
* “**Chức vụ**” có kiểu dữ liệu là Text.

Như đã nói trước đó, bảng User sẽ lấy “**Email**" làm khoá chính nhưng ta sẽ chọn “**Tên NV**” là biến hiển thị trên ứng dụng bằng cách bỏ tích chọn “**Email**” và chọn tích chọn “**Tên NV**” trong mục LABEL?. Tiếp đến ta phải có các ràng buộc cho bảng này:

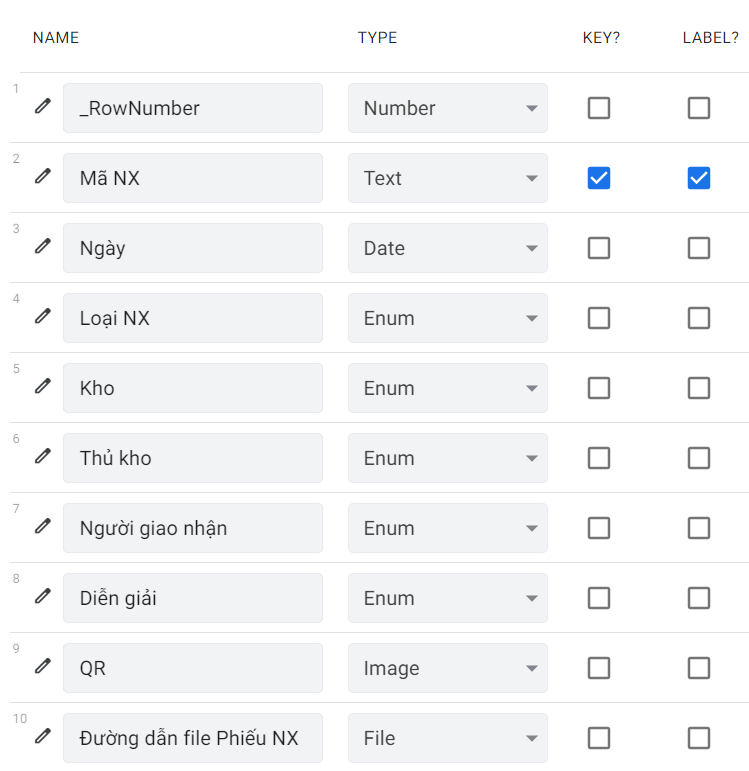


*Hình 3.29. Ràng buộc trong bảng User*

Trong mục REQUIRE?, có 2 trường dữ liệu bắt buộc phải có đó là “**Email**” và “**SĐT**”. Mỗi một cá nhân đều có một email và số điện thoại không trùng nhau, chính vì vậy 2 trường dữ liệu này là bắt buộc phải nhập. Ngoài ra đây cũng là 2 phương thức liên lạc thường thấy nhất hiện nay.

Như vậy đối với bảng User ta đã thiết lập xong. Bây giờ ta sẽ tiếp tục với bảng Phieu NX.

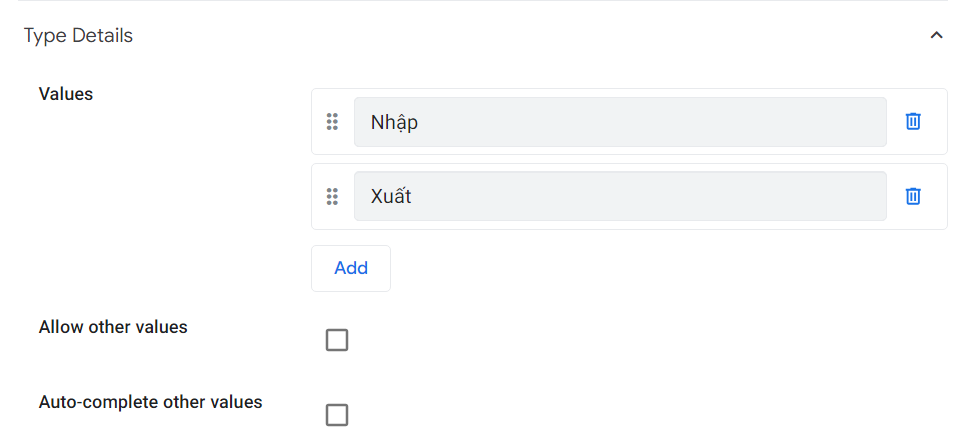
#### **Điều chỉnh bảng Phieu NX**



*Hình 3.30. Các trường dữ liệu trong bảng Phieu NX*

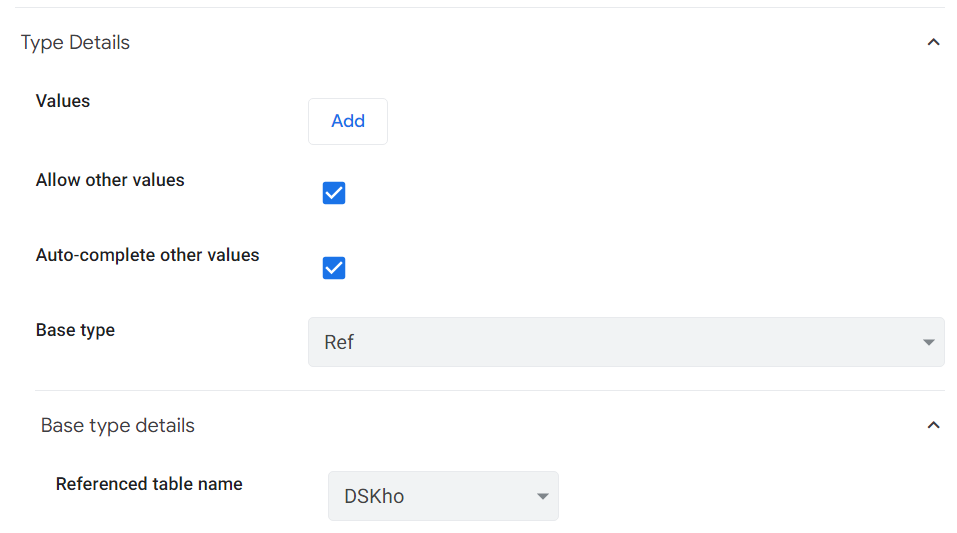
Trong bảng Phieu NX ta sẽ thiết lập các kiểu dữ liệu như hình 3.30, cụ thể như sau:

* “**Ma NX**” sẽ có kiểu dữ liệu là Text. Trong bảng dữ liệu này, “**Ma NX**” sẽ là khoá chính của bảng.
* “**Ngày**” có kiểu dữ liệu là Date.
* “**Loại NX**” có kiểu dữ liệu là Enum. Kiểu dữ liệu Enum sẽ cho ra một danh sách có sẵn hoặc người dùng có thể tự thêm vào. Trong đó ta sẽ chỉ thêm 2 loại dữ liệu là Nhập hoặc Xuất như hình 3.31. Để set 2 kiểu dữ liệu là duy nhất và không được thêm hay chỉnh sửa từ phía người dùng, trong phần Edit > Type Details ta thêm 2 kiểu dữ liệu là “Nhập” và “Xuất”, sau đó ta bỏ tích chọn Allow other values và Auto-complete other values.



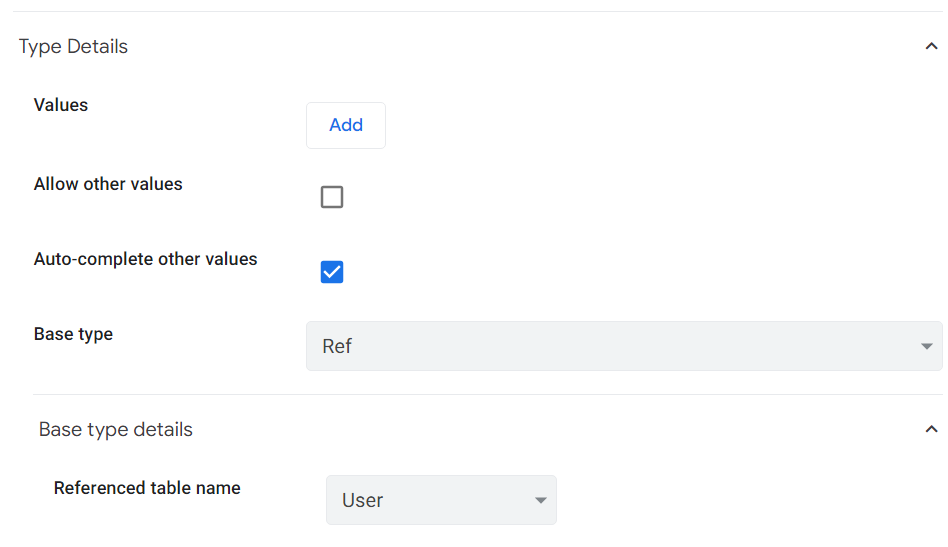
*Hình 3.31. 2 Kiểu dữ liệu trong Loại NX*

* “**Kho**” có kiểu dữ liệu là Enum. Để có thể lấy được danh sách kho từ bảng DSKho, trong mục Edit > Type Details > Base type ta chọn kiểu là Ref và chọn Referenced table name là DSKho. Như vậy khi người dùng chọn dữ liệu từ trường “**Kho**” thì sẽ có một danh sách kho đổ từ bảng dữ liệu DSKho.



*Hình 3.32. Kho sẽ lấy dữ liệu từ bảng DSKho*

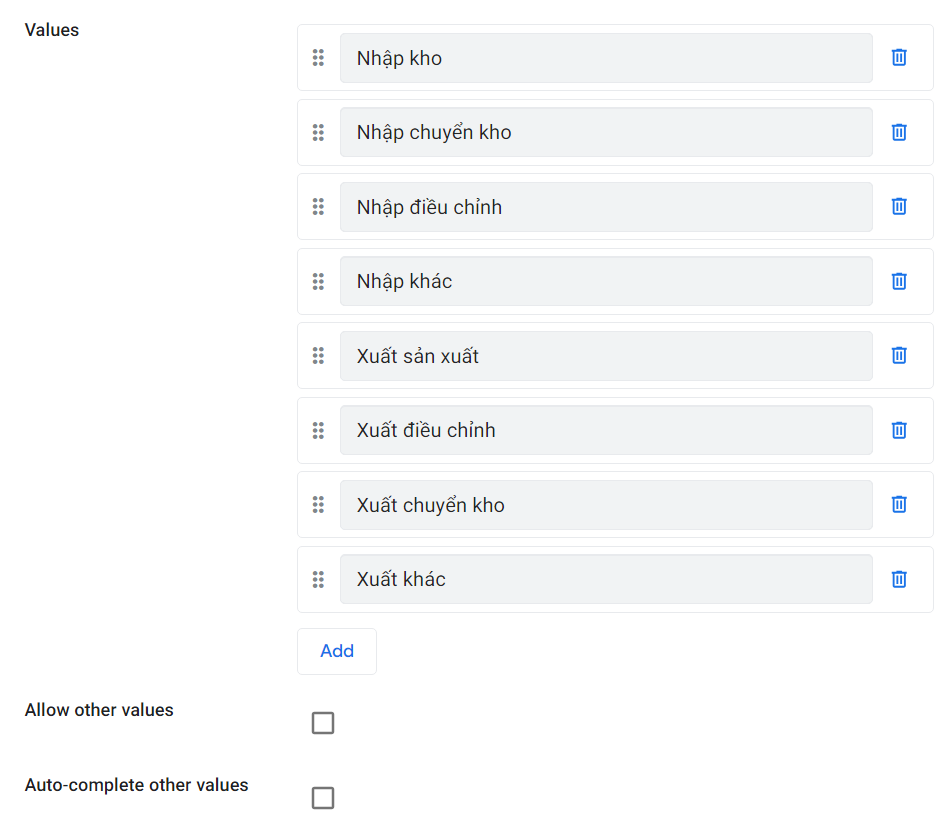
* “**Thủ kho**” có kiểu dữ liệu là Enum. Tương tự như trên “**Thủ kho**” sẽ lấy dữ liệu từ bảng User để khi người dùng chọn thủ kho sẽ có tên người dùng đã lưu sẵn xuất hiện.



*Hình 3.33. Thủ kho sẽ lấy dữ liệu từ bảng User*

* “**Người giao nhận**” có kiểu dữ liệu là Enum
* “**Diễn giải**” có kiểu dữ liệu là Enum. Trong đó ta sẽ thêm các loại dữ liệu để miêu tả hành động lên phiếu nhập xuất:
* Nhập kho
* Nhập chuyển kho
* Nhập điều chỉnh
* Nhập Khác
* Xuất sản xuất
* Xuất điều chỉnh
* Xuất chuyển kho
* Xuất khác

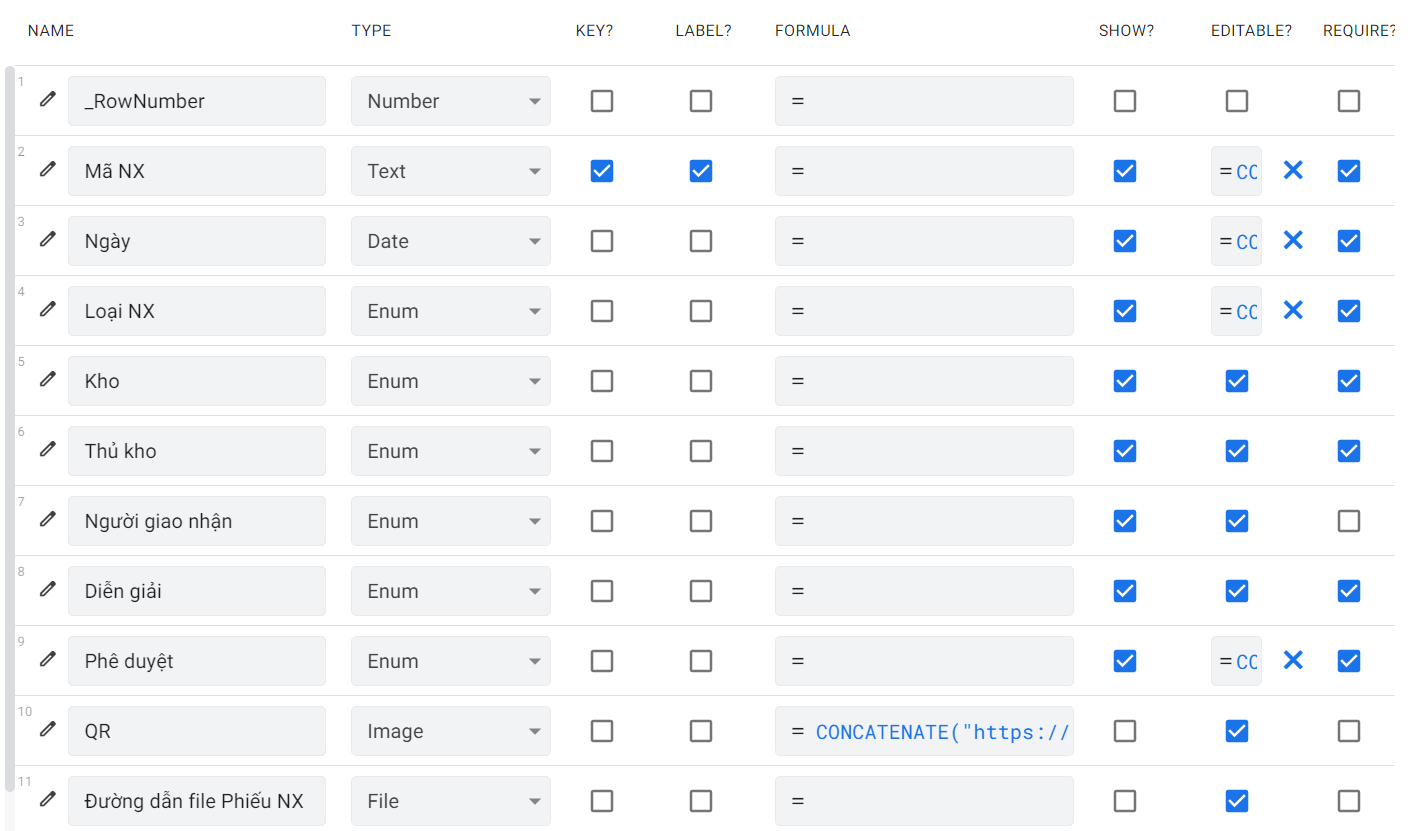
Để kiểu dữ liệu này là duy nhất và không được thêm hay chỉnh sửa từ phía người dùng, trong phần Edit > Type Details ta thêm các kiểu dữ liệu như hình 3.34, sau đó ta bỏ tích chọn Allow other values và Auto-complete other values.



*Hình 3.34. Cấc kiểu dữ liệu trong Diễn giải*

* “**QR**” có kiểu dữ liệu là Image. Đối với “**QR**” được thiết lập kiểu dữ liệu Image là để chứa hình ảnh QR code của phiếu nhập.
* “**Đường dẫn file Phiếu NX**“ có kiểu dữ liệu là File. Còn đối với “**Đường dẫn file Phiếu NX**” sẽ có kiểu dữ liệu là File vì trong các phần sau trường dữ liệu này sẽ là nơi lưu trữ các đường dần file PDF chứa thông tin phiếu nhập xuất của ứng dụng.

Tiếp đến trong bảng này ta sẽ đi thiết lập các ràng buộc:



*Hình 3.35. Ràng buộc trong bảng Phieu NX*

Trừ “**Người giao nhận**”, “**QR**“ với “**Đường dẫn file Phiếu NX**” ra thì tất cả những trường dữ liệu còn lại đều bắt buộc phải nhập. Sẽ có 3 trường dữ liệu đặc biệt cần thiết lập ràng buộc kĩ là “**Mã NX**”, “**Ngày**” và “**Loại NX**”.

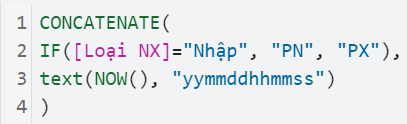
Đối với “**Mã NX**”, “**Ngày**”, “**Loại NX**” ta sẽ thêm một công thức trong mục EDITABLE? như sau:



*Hình 3.36. Công thức không cho phép người dùng chỉnh sửa dữ liệu*

Đây là công thức thể hiện rằng cả 3 trường dữ liệu này là ứng dụng từ tạo và người dùng không được phép chỉnh sửa gì.

Các ràng buộc tiếp theo sẽ được điều chỉnh riêng cho từng trường dữ liệu. Đầu tiên là “**Ma NX**”:



*Hình 3.37. Công thức tự động đặt tên phiếu nhập xuất*

Ở mục INITIAL VALUE, ta sẽ nhập công thức như hình 3.37. Hàm CONCATENATE() để ghép các kiểu dữ liệu Text lại với nhau. Nếu như “**Loại NX**” là Nhập thì tên phiếu sẽ bắt đầu bẳng kí tự PN ngược lại sẽ là PX tương ứng cho phiếu xuất. Ta dùng hàm TEXT(NOW()) để để lấy thời gian thực tế để tạo phiếu, định dạng sẽ là yymmddhhmmss. Ví dụ: Nếu ta tạo 1 phiếu xuất lúc 23 giờ 43 phút 34 giây, ngày 24 tháng 4 năm 2024 thì tên phiếu nhập lúc này sẽ tự động điền là PX240424234334.

Cũng ở mục INITIAL VALUE, trường dữ liệu “**Ngày**” ta sẽ dùng hàm TODAY() để lấy ngày thực tế để hiện trên ứng dụng:



*Hình 3.38. Hàm TODAY()*

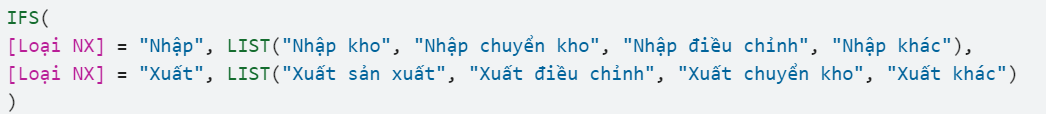
Tiếp theo, dữ liệu “**Loại NX**” ta sẽ dùng công thức như hình dưới:



*Hình 3.39. Công thức phân biệt loại xuất nhập*

Công thức này để phân biệt rằng nếu ta chọn chức năng nhập kho thì ”**Loại NX**” sẽ tự điền cho chúng ta là Nhập.

Đối với trường dữ liệu “**Diễn giải**”, vì đã thêm 8 dữ liệu khác nhau nên ta gặp một vấn đề đó là phiếu xuất hay nhập gì ứng dụng cũng sẽ hiển thị 8 lựa chọn cho người dùng. Để tách ra khi nào là phiếu nhập thì chỉ xuất hiện các lựa chọn cho phiếu nhập ta làm theo các bước vào Edit chọn Data Validity > Valid if và nhập công thức như sau:



*Hình 3.40. Công thức phân biệt các lựa chọn diễn giải cho từng loại phiếu*

Trường dữ liệu “**QR**” như đã nêu ở trên là dùng để chứa hình ảnh QR code. Qr code này ta sẽ dùng API của bên thứ ba để tạo và công thức như sau:



*Hình 3.41. Công thức dùng API của bên thứ 3 để tạo QR Code*

API này sẽ tạo ra một QR code với kích thước là 100px x 100px mà khi quét vào sẽ ra mã nhập xuất:



*Hình 3.42. Kết quả khi quét QR Code trên điện thoại*

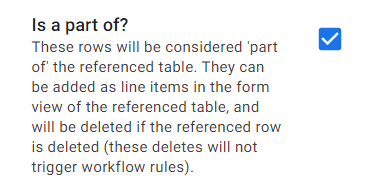
#### **Điều chỉnh bảng PhieuNXChitiet**



*Hình 3.43. Các trường dữ liệu trong bảng PhieuNXChitiet*

Trong bảng PhieuNXChitiet ta sẽ thiết lập các kiểu dữ liệu như hình 3.43, ta có:

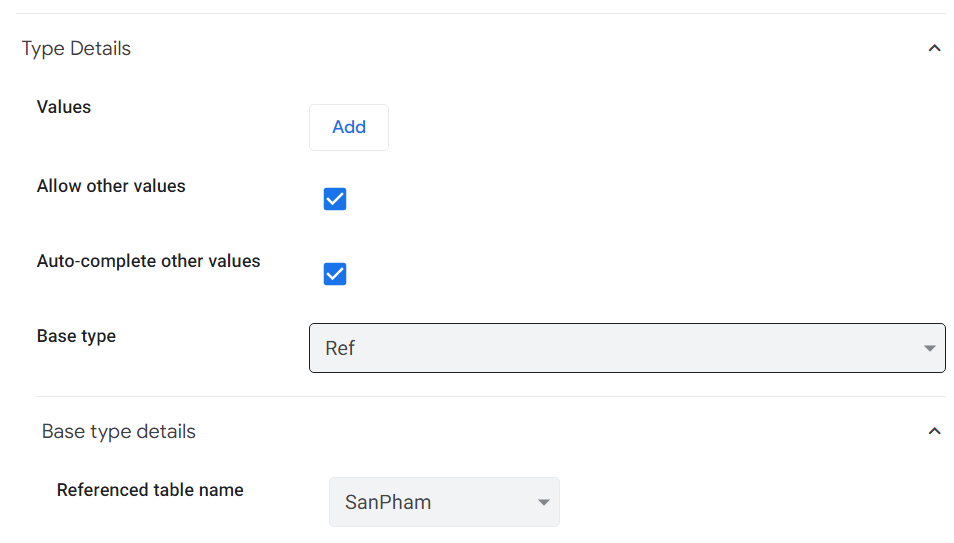
* “**id**" sẽ có kiểu dữ liệu là Text. Đây cũng sẽ là khoá chính của bảng này. “**id**” này ta sẽ để Appsheet tự động nhập ngẫu nhiên.
* “**Mã NX**” có kiểu dữ liệu là Ref. Đây là khoá ngoại của bảng nối với bảng Phieu NX để lấy dữ liệu từ đó. Tuy nhiên ta cần làm thêm 1 bước nhỏ nữa để có thể nhập dữ liệu. Trong mục Edit > Type Details > Is a part of? ta tích chọn mục này.



*Hình 3.44. Chọn Is a part of?*

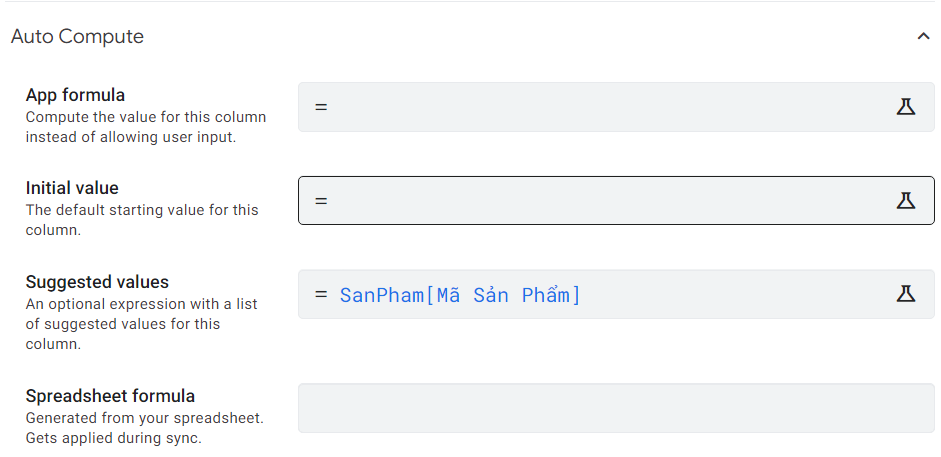
Việc chọn mục này có ý nghĩa ràng khi ta xoá dữ liệu trong bảng Phieu NX thì các dữ liệu trong bảng PhieuNXChitiet cũng sẽ bị xoá.

* “**Sản phẩm**” có kiểu dữ liệu là Enum. Trong mục Edit > Type Details > Base type ta chọn kiểu là Ref và chọn Referenced table name là SanPham. Như vậy khi người dùng chọn dữ liệu từ trường “**Sản phẩm**” thì sẽ có một danh sách sản phẩm đổ từ bảng dữ liệu SanPham.



*Hình 3.45. Sản phẩm lấy dữ liệu từ bảng SanPham*

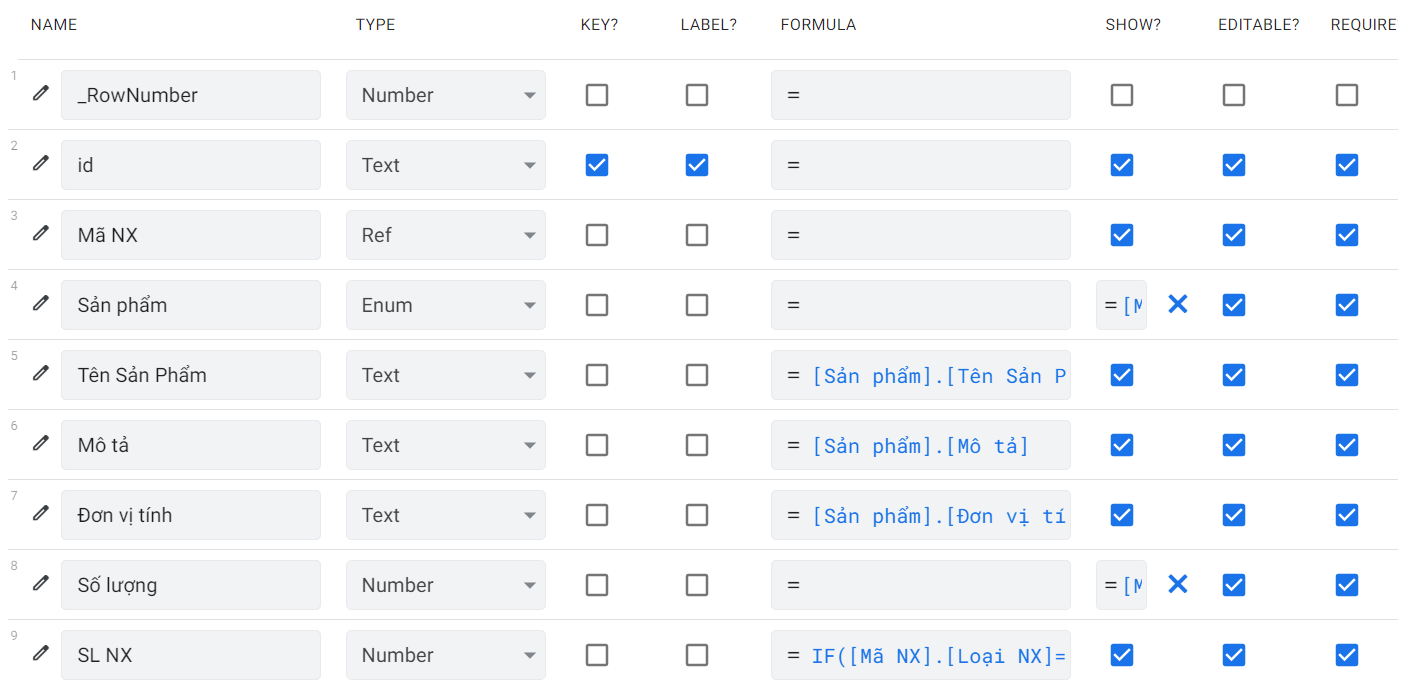
Tiếp theo trong mục Edit > Auto Compute > Suggested values là nhập như hình dưới để bảng PhieuNXChitiet có thể đọc được thông tin sản phẩm từ bảng SanPham.



*Hình 3.46. Nhập điều kiện để có thể đọc dữ liệu từ bảng SanPham*

* “**Tên Sản Phẩm**” có kiểu dữ liệu là Text.
* “**Mô tả**” có kiểu dữ liệu là Text.
* “**Đơn vị tính**” có kiểu dữ liệu là Text.
* “**Số lượng**” có kiểu dữ liệu là Number.
* “**SL NX**” có kiểu dữ liệu là Number.

Tiếp theo ta sẽ đi thiết lập các ràng buộc. Như hình 3.47, ta có thể thấy tất cả các dữ liệu của bảng ở mục REQUIRE? Là bắt buộc phải nhập.



*Hình 3.47. Ràng buộc trong bảng PhieuNXChitiet*

Bắt đầu từ “**Tên Sản Phẩm**”, “**Mô tả**” và “**Đơn vị tính**” ta sẽ dùng tham chiếu tới bảng SanPham thông qua công thức như các hình 3.48, hình 3.49 và hình 3.50.



*Hình 3.48. Tham chiếu đến cột Tên Sản Phẩm trong bảng SanPham*



*Hình 3.49. Tham chiếu đến cột Mô tả trong bảng SanPham*



*Hình 3.50. Tham chiếu đến cột Đơn vị tính trong bảng SanPham*

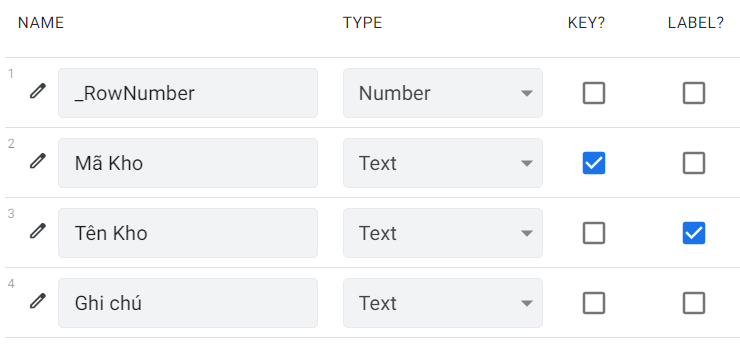
Cuối cùng đối với “**SL NX**” ta cũng thêm một công thức để hiện thị trạng thái nhập xuất trên ứng dụng.



*Hình 3.51. Công thức SL NX*

Giải thích cho công thức trên như sau: nếu trong kho ta nhập vào số lượng là 10 thì khi xuất kho số lượng sẽ phải -10. Việc thêm cột này trong bảng để về sau khi ta kiểm kê số lượng nhập – xuất – tồn sẽ nhanh hơn và tường minh hơn.

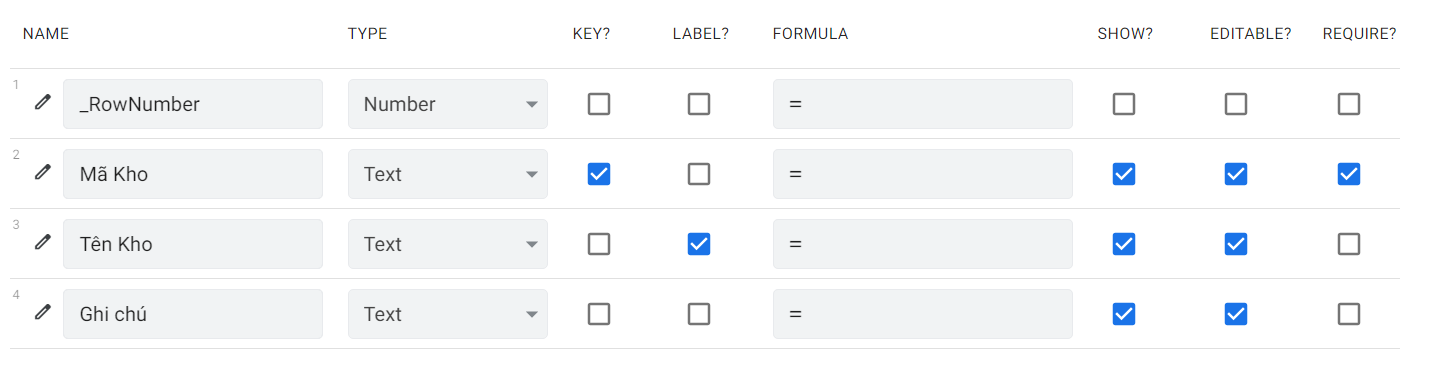
#### **Điều chình bảng DSKho**



*Hình 3.52. Các trường dữ liệu trong bảng DSKho*

Trong bảng DSKho:

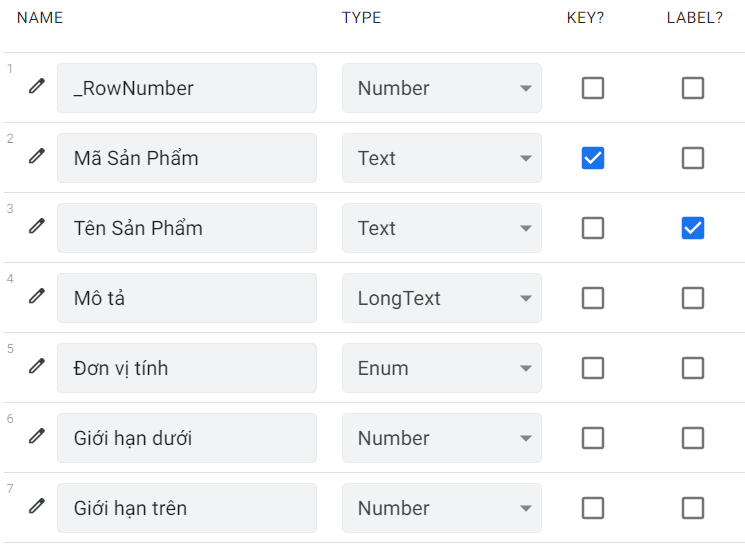
* “**Mã Kho**” có kiểu dữ liệu là Text. Đây sẽ là khoá chính của bảng DSKho.
* “**Tên Kho**” có kiểu dữ liệu là Text. Đây sẽ là dữ liệu dùng để để hiển thị trên ứng dụng.
* “**Ghi chú**” có kiểu dữ liệu là Text.



*Hình 3.53. Ràng buộc trong bảng DSKho*

Ở mục REQUIRE? bảng dữ liệu này chỉ có duy nhất “**Mã Kho**” là yêu cầu phải nhập.

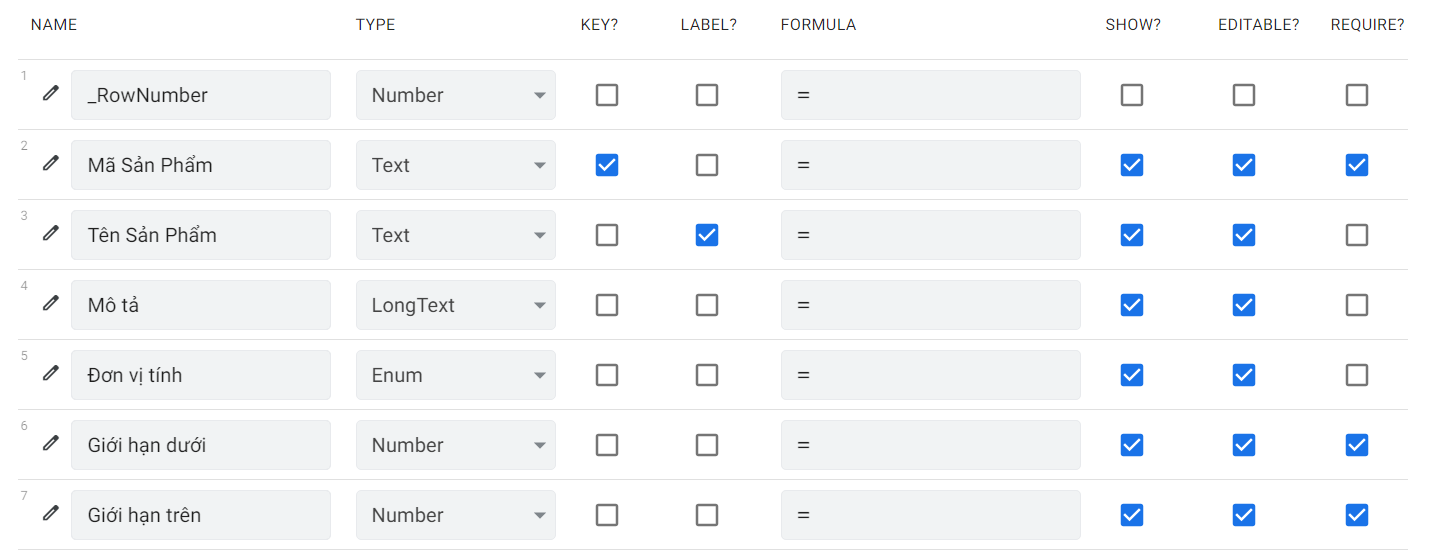
#### **Điều chỉnh bảng SanPham**



*Hình 3.54. Các trường dữ liệu trong bảng SanPham*

Trong bảng SanPham:

* “**Mã Sản Phẩm**” có kiểu dữ liệu là Text. Đây là khoá chính của bảng SanPham.
* “**Tên Sản Phẩm**” có kiểu dữ liệu là Text. Đây sẽ là dữ liệu dùng để để hiển thị trên ứng dụng.
* “**Mô tả**” có kiểu dữ liệu là LongText.
* “**Đơn vị tính**” có kiểu dữ liệu là Enum.
* “**Giới hạn dưới**” có kiểu dữ liệu là Number.
* “**Giới hạn trên**” có kiểu dữ liệu là Number.

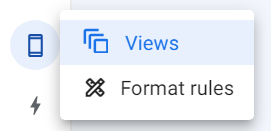


*Hình 3.55. Ràng buộc trong bảng SanPham*

Ở mục REQUIRE? bảng dữ liệu này yêu cầu không được để trống dữ liệu của trường “**Mã Sản Phẩm**”, “**Giới hạn dưới**” và “**Giới hạn trên**”.

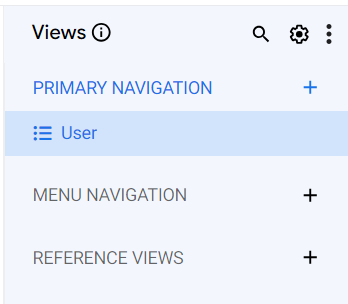
### **Dùng view để tạo giao diện cho ứng dụng**

Sau khi ta thiết lập và ràng buộc các dữ liệu kĩ càng như trên, bây giờ ứng dụng của chúng ta đã sẵn sàng để sử dụng. Công việc bây giờ là sẽ đi tạo giao diện cho các chức năng trong ứng dụng. Đầu tiên ta vào Views:



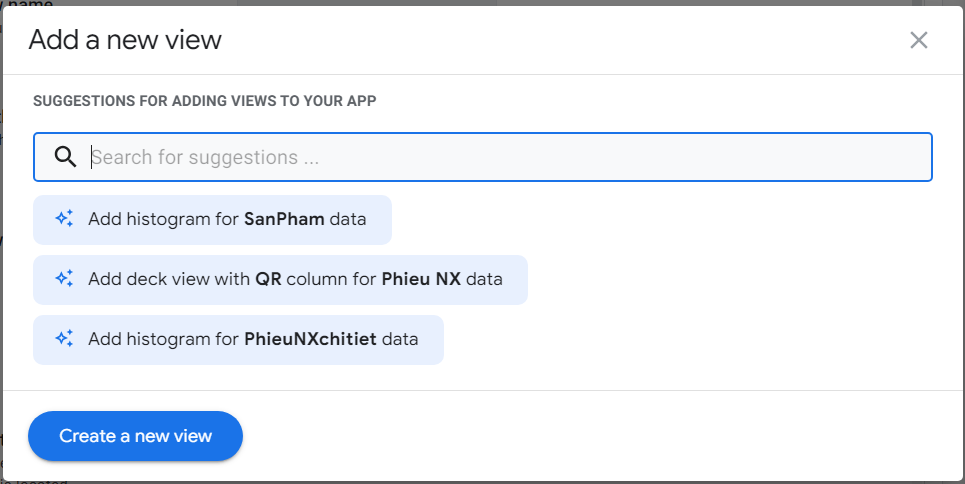
*Hình 3.56. Vào Views*

Trong View ta sẽ có danh sách các button hiện trên ứng dụng như sau:



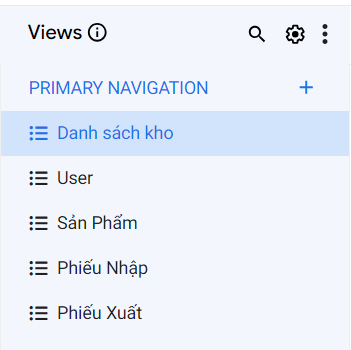
*Hình 3.57. Nơi chức các chức năng của ứng dụng*

Để thuận tiện cho việc tạo giao diện, ta sẽ thêm tất cả các views cần thiết cho ứng dụng trước. Ta bấm vào dấu “**+**” trên thanh PRIMARY NAVIGATION để thêm các views.



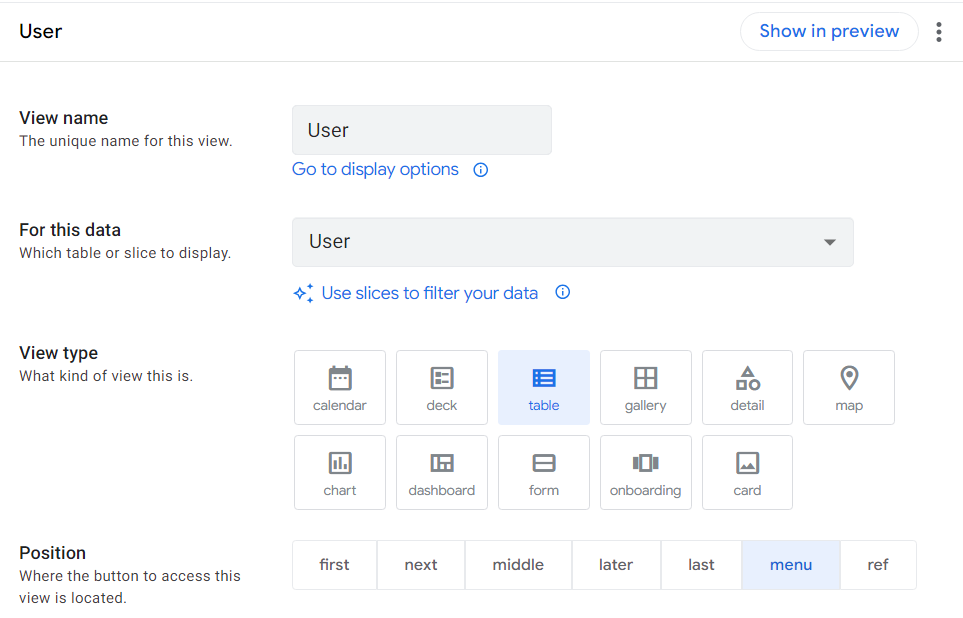
*Hình 3.58. Thêm Views mới*

Chọn Create a new view. Sau khi có được danh sách như hình 3. ta sẽ bắt đầu chỉnh view cho từng chức năng.



*Hình 3.59. Danh sách các Views sẽ được chỉnh sửa*

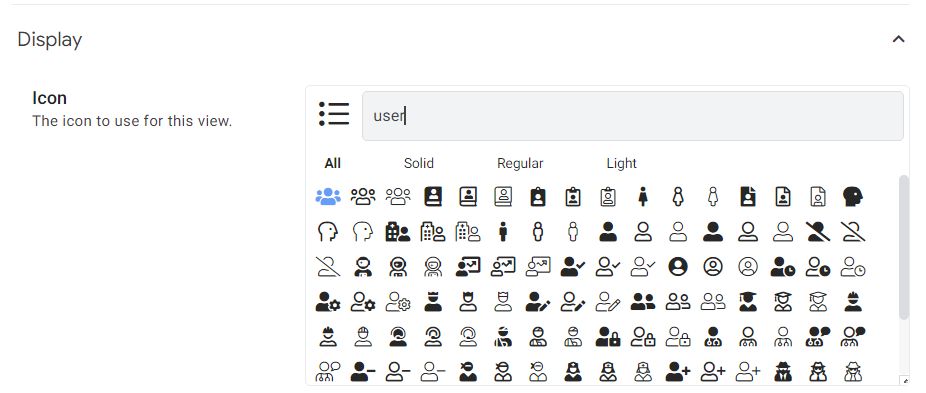
#### **Views cho User**



*Hình 3.60. Views của User*

Trong phần views cho User:

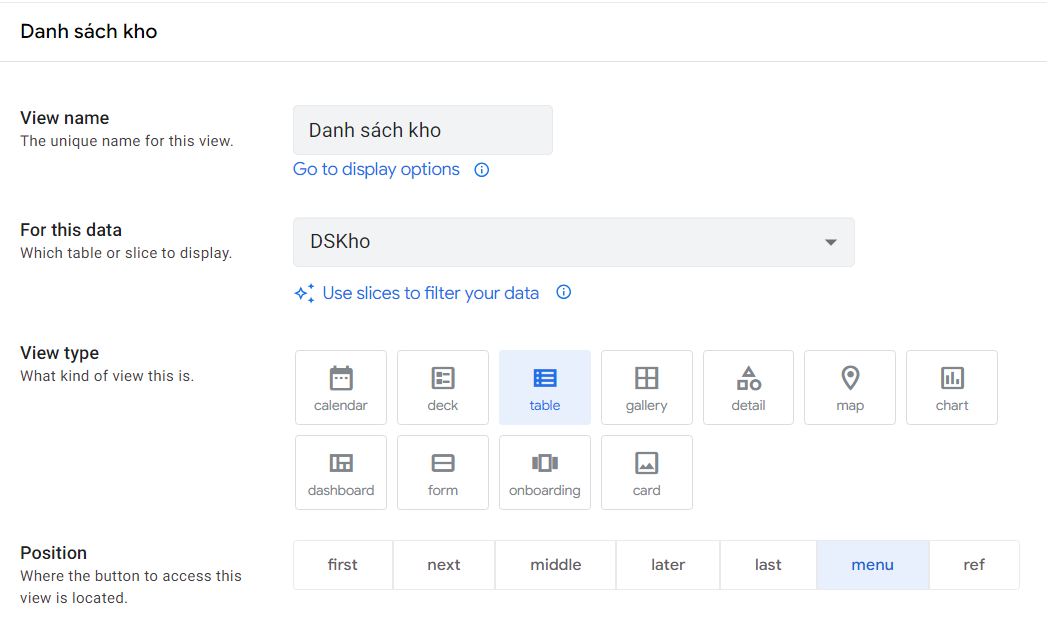
* View name ta sẽ đặt là User.
* For this data ta sẽ dùng dữ liệu từ bảng User.
* View type ta sẽ để là kiểu table.
* Position ta sẽ chọn là menu.



*Hình 3.61. Chọn icon cho User*

Ở bên dưới, mục Display > Icon ta sẽ chọn icon cho chức năng này.

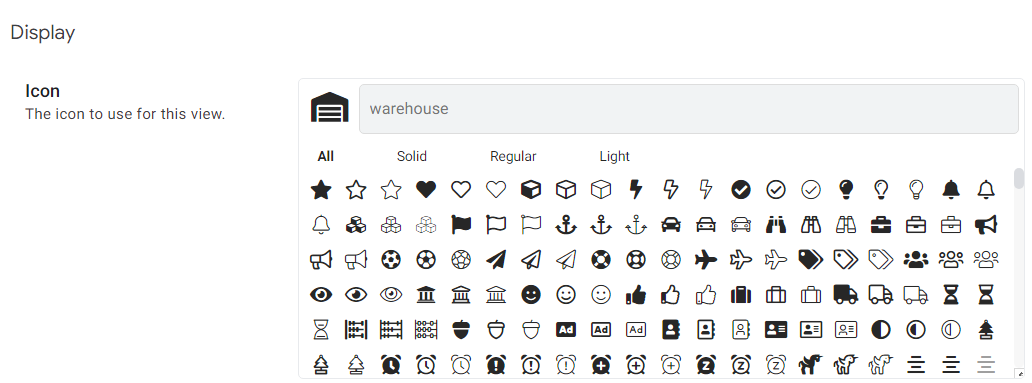
#### **Views cho Danh sách kho**



*Hình 3.62. Views của Danh sách kho*

Trong phần views cho Danh sách kho:

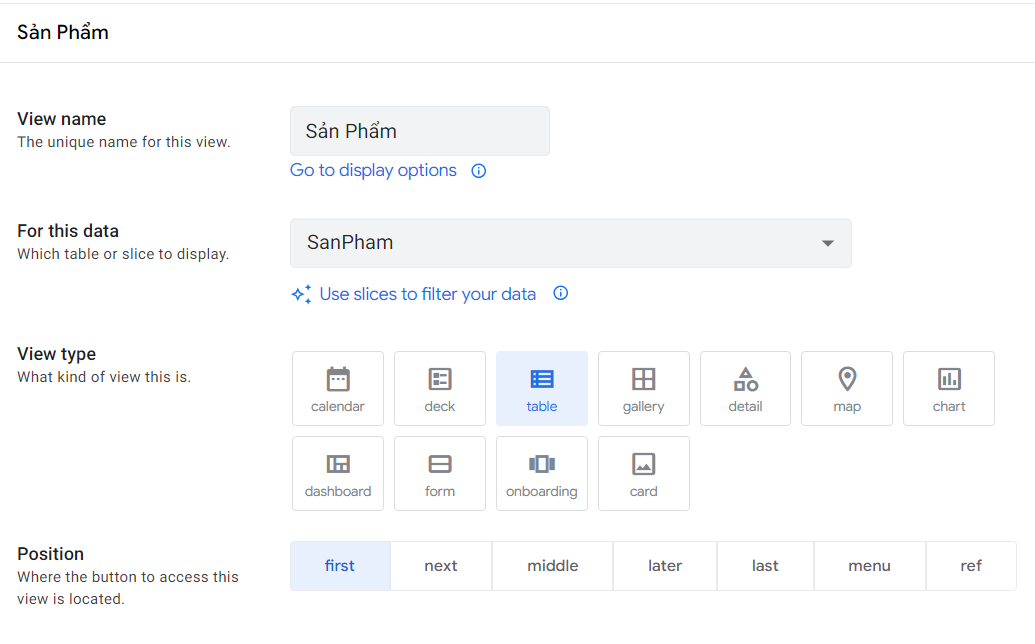
* View name ta sẽ đặt là Danh sách kho.
* For this data ta sẽ dùng dữ liệu từ bảng DSKho.
* View type ta sẽ để là kiểu table.
* Position ta sẽ chọn là menu.



*Hình 3.63. Chọn icon cho Danh sách kho*

Ở bên dưới, mục Display > Icon ta sẽ chọn icon cho chức năng này.

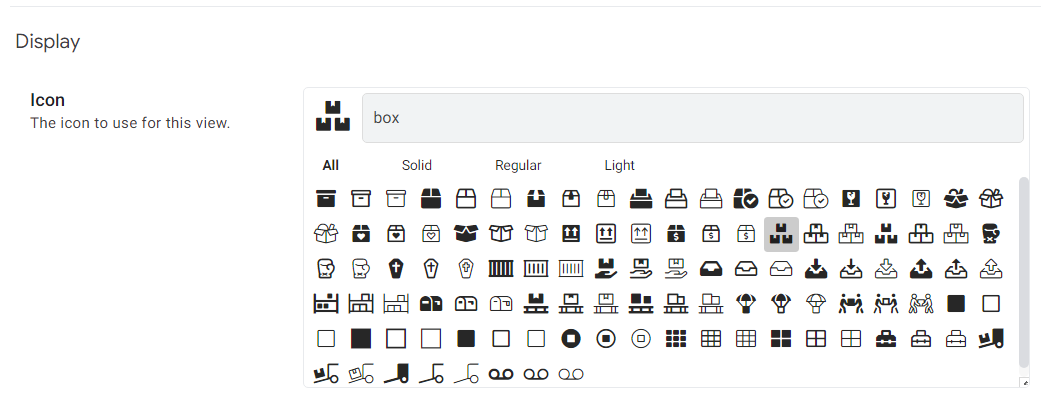
#### **Views cho Sản Phẩm**



*Hình 3.64. Views của Sản Phẩm*

Trong phần views cho Sản Phẩm:

* View name ta sẽ đặt là Sản Phẩm.
* For this data ta sẽ dùng dữ liệu từ bảng SanPham.
* View type ta sẽ để là kiểu table.
* Position ta sẽ chọn là first.

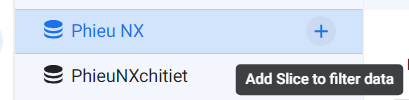


*Hình 3.65. Chọn icon cho Sản Phẩm*

Ở bên dưới, mục Display > Icon ta sẽ chọn icon cho chức năng này.

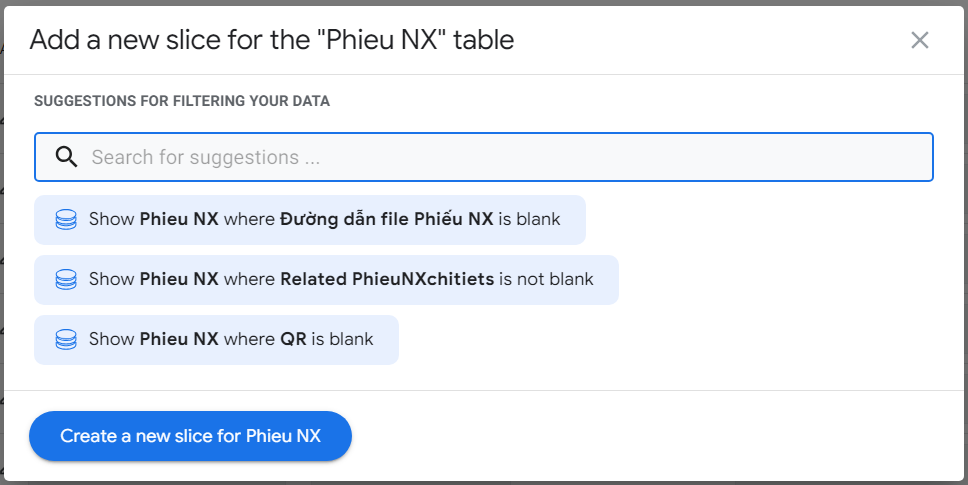
#### **Views cho Phiếu Nhập**

Để tạo view cho Phiếu Nhập, ta phải quay lại Data > Add Slice to filter data.



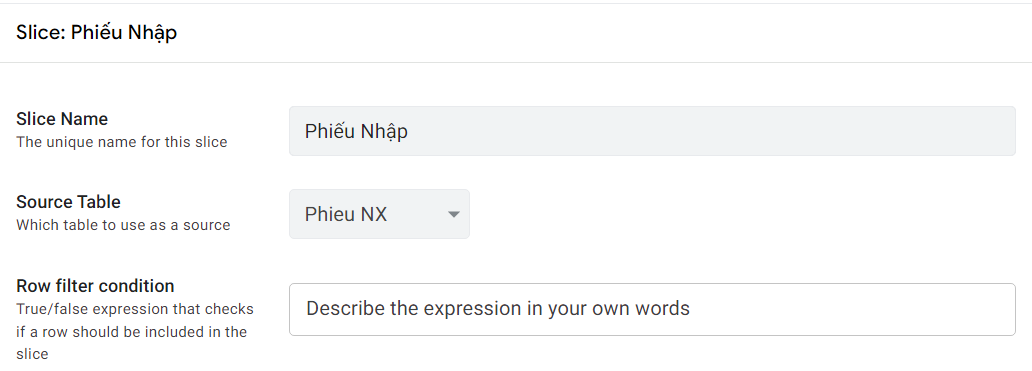
*Hình 3.66. Tạo Slice Phiếu Nhập*

Vì ta chỉ có phiếu nhập xuất chứ không có riêng lẻ từng phiếu nên bước này để ta có thể tách Phieu NX ra làm đôi nhưng vẫn có thể dùng chung cơ sở dữ liệu.



*Hình 3.67. Tạo 1 Slice mới*

Ta chọn Create a new slide for Phieu NX.



*Hình 3.68. Chỉnh sửa thông tin slice Phiếu Nhập*

Trong Slice: Phiếu Nhập:

* Slice Name ta đặt tên là Phiếu Nhập.
* Source Table ta sẽ lấy từ bảng Phieu NX.
* Row filter condition ta chọn Create a custom expression



*Hình 3.69. Chọn Create a custom expression*

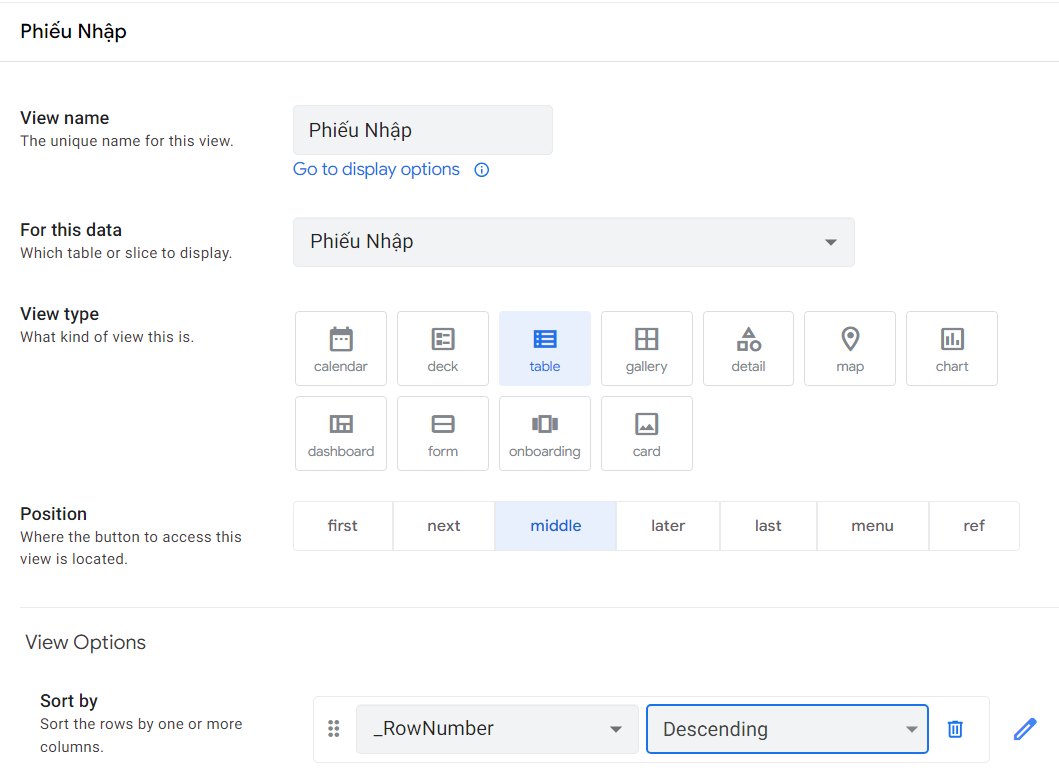
Trong đó ta sẽ nhập công thức như sau:



*Hình 3.70. Công thức trong slice Phiếu Nhập*

Như vậy ta đã tạo thành công 1 slice dành cho Phiếu Nhập.

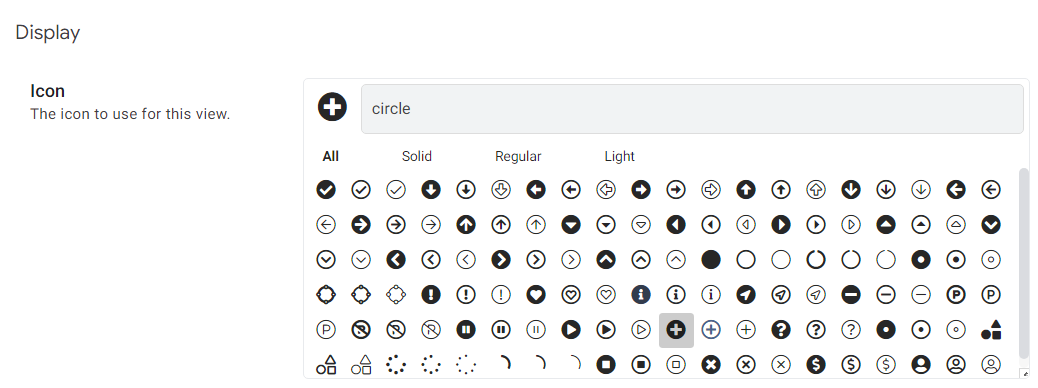
Quay lại với Views, ta chỉnh sửa như sau:



*Hình 3.71. Views của Phiếu Nhập*

Trong phần views cho Phiếu Nhập:

* View name ta sẽ đặt là Phiếu Nhập.
* For this data ta sẽ dùng dữ liệu từ Silce: Phiếu Nhập.
* View type ta sẽ để là kiểu table.
* Position ta sẽ chọn là middle.
* View Options > Sort by ta đặt \_RowNumber – Descending.

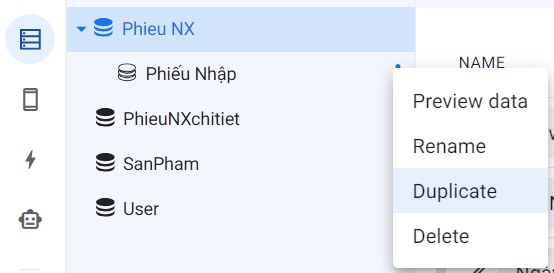


*Hình 3.72. Chọn icon cho Phiếu Nhập*

Ở bên dưới, mục Display > Icon ta sẽ chọn icon cho chức năng này.

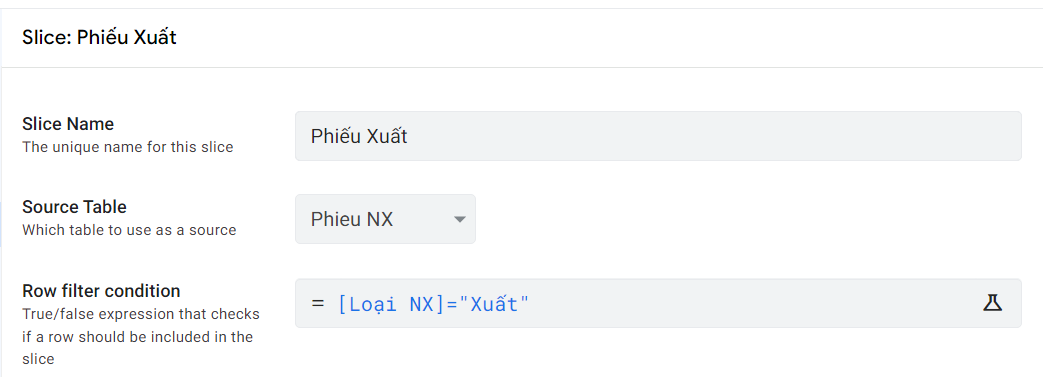
#### **Views cho Phiếu Xuất**

Tương tự như Phiếu Nhập, Phiếu cũng sẽ phải tạo 1 slice để có thể tách biệt với Phiếu Nhập. Để tiết kiệm thời gian ta sẽ Duplicate Phiếu Nhập và sửa các dữ liệu nhập thành xuất.



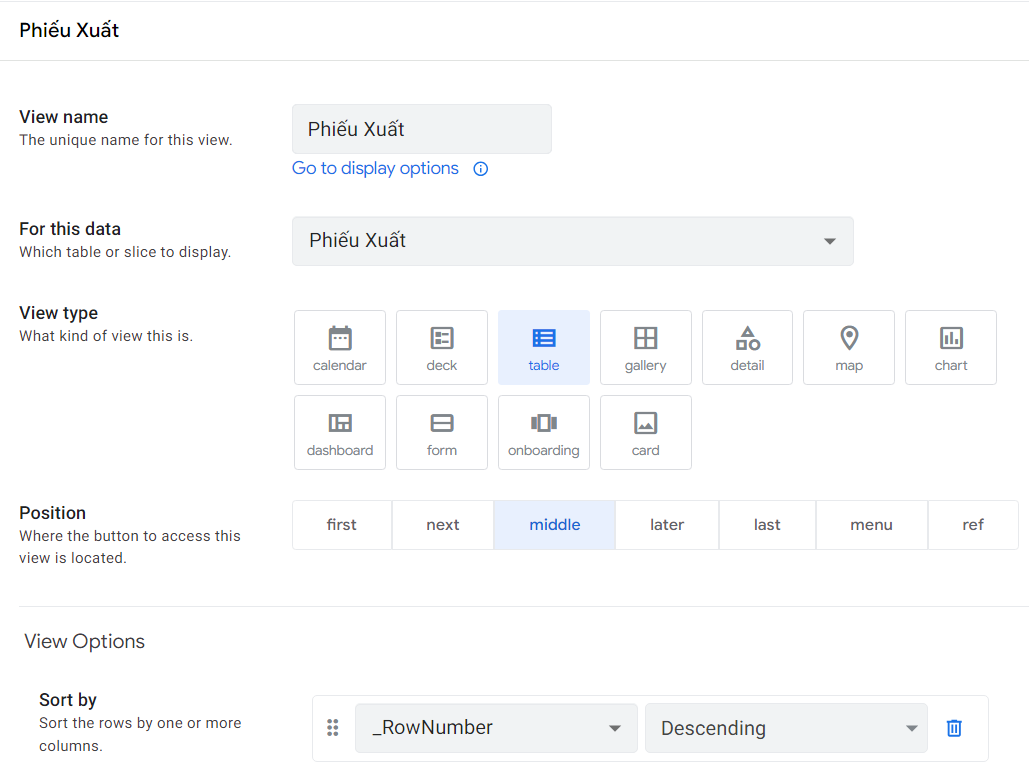
*Hình 3.73. Duplicate Phiếu Nhập để tạo nhanh Phiếu Xuất*

Thông tin được sửa lại sẽ như hình sau:



*Hình 3.74. Chỉnh sửa thông tin của Slide: Phiếu Xuất*

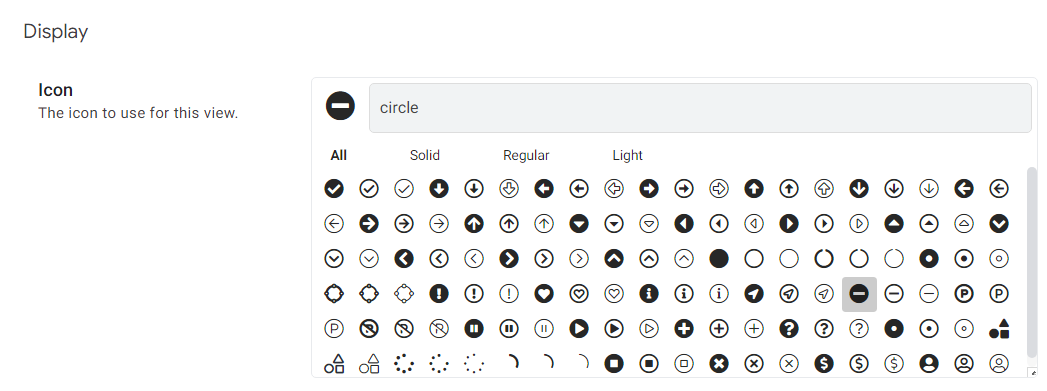
Như vậy là ta đã có 1 Slice: Phiếu Xuất hoàn chỉnh. Tiếp theo ta quay lại Views để tạo giao diện cho Phiếu Xuất.



*Hình 3.75. Views của Phiếu Xuất*

Trong phần views cho Phiếu Xuất:

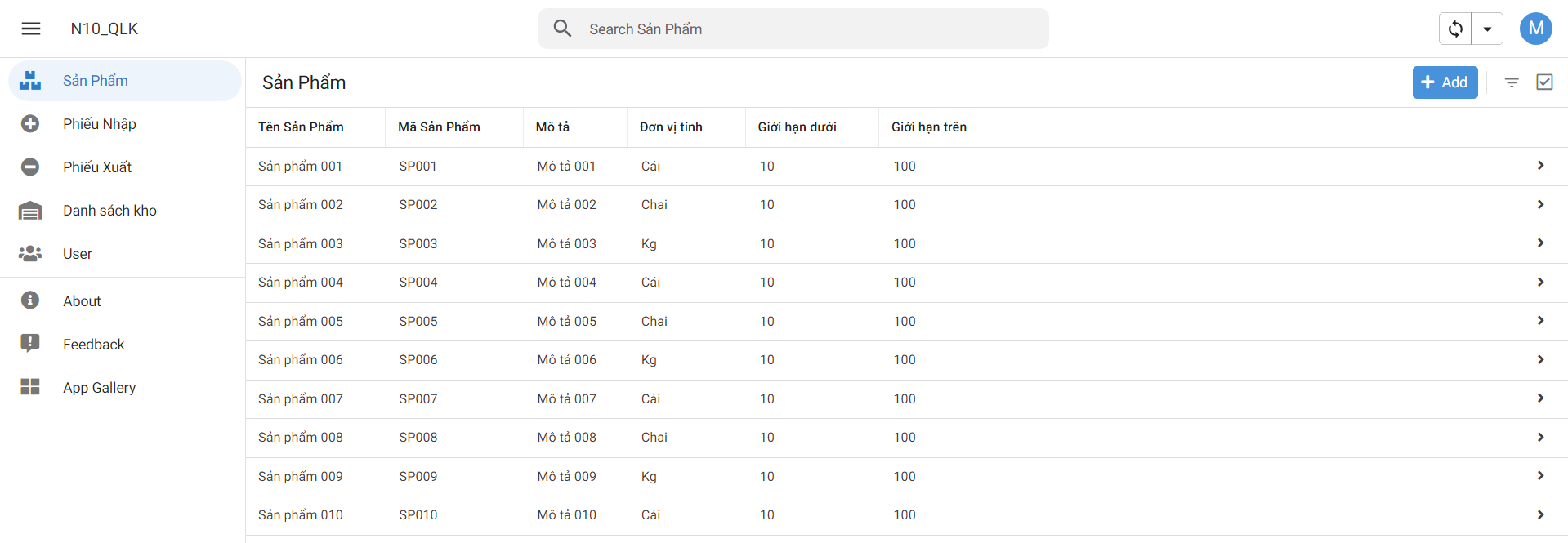
* View name ta sẽ đặt là Phiếu Xuất.
* For this data ta sẽ dùng dữ liệu từ Silce: Phiếu Xuất.
* View type ta sẽ để là kiểu table.
* Position ta sẽ chọn là middle.
* View Options > Sort by ta đặt \_RowNumber – Descending.



*Hình 3.76. Chọn icon cho Phiếu Xuất*

Ở bên dưới, mục Display > Icon ta sẽ chọn icon cho chức năng này.

Kết quả đạt được sau khi hoàn thành tất cả bước trên như sau:



*Hình 3.77. Giao diện ứng dụng với đầy đủ chức năng*

## **Chức năng** **xuất phiếu nhập xuất kho**

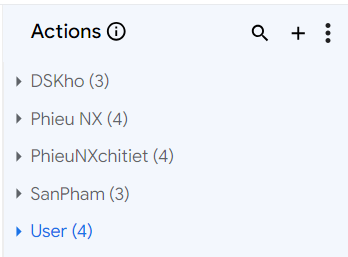
### **Tạo Actions**

Để xuất được file PDF từ Appsheet ta sẽ dùng Actions của Appsheet.



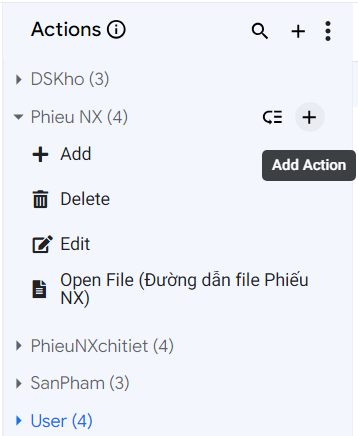
*Hình 3.78. Actions*

Để tạo actions đầu tiên ta phải chọn bảng dữ liệu cần tạo actions



*Hình 3.79. Danh sách các dữ liệu có thể tạo Actions*

Sau đó chọn Add Action để khởi tạo 1 action mới trong bảng.



*Hình 3.80. Tạo action mới*

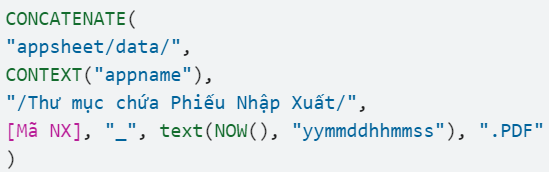
Ta sẽ add tổng cộng là 4 actions thay vì 5 bởi vì đã có sẵn 1 action Open File:

* Tạo đường dẫn file Phiếu NX



*Hình 3.81. Action Tạo đường dẫn file Phiếu NX*

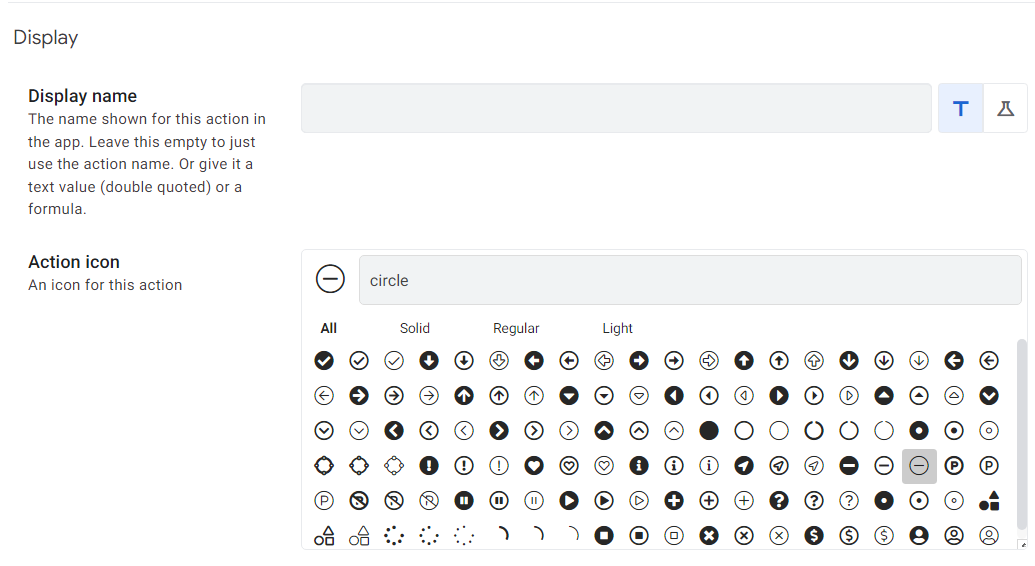
* Action name ta đặt là Tạo đường dẫn file Phiếu NX.
* For a record of this table ta sẽ dùng dữ liệu từ bảng Phieu NX.
* Do this ta sẽ chọn Data: set the values of some columns in this row.
* Set these columns ta chọn cột Đường dẫn file Phiếu NX và đặt công thức như sau:



*Hình 3.82. Công thức tạo đường dẫn file PDF*

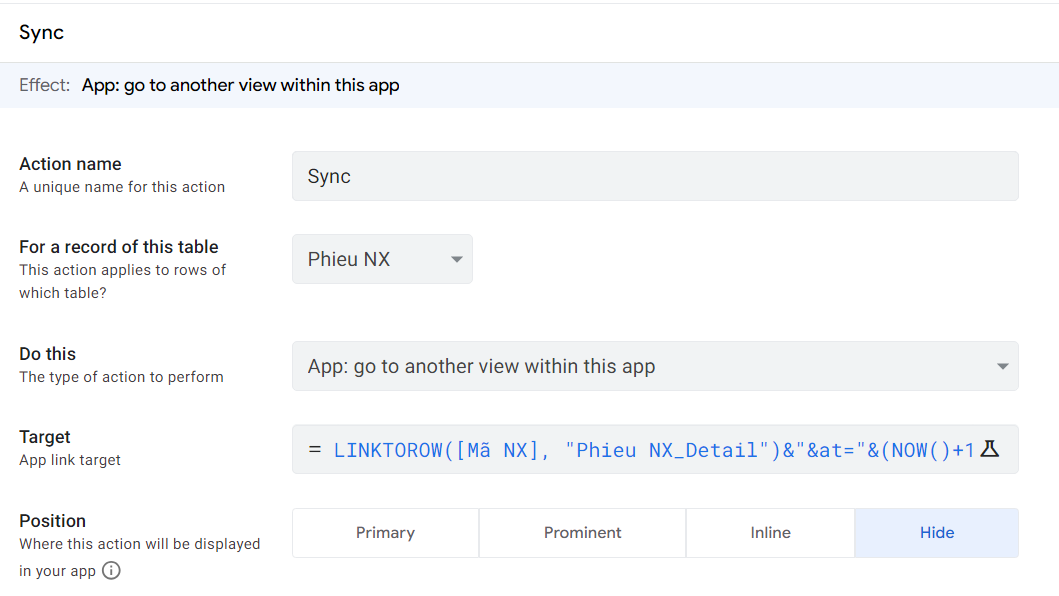
Công thức này sẽ tạo 1 đường dẫn đến thư mục có đường dẫn như sau: appsheet/data/Tên của ứng dụng/Thư mục chứa Phiếu Nhập Xuất/. Tên của file sẽ dùng thời gian tạo phiếu để làm tên với định đạng yymmddhhmmss.

* Position ta chọn Hide vì action này sẽ không hiển thị trên ứng dụng.
* Vào mục Display > Action icon để chọn icon cho action.



*Hình 3.83. Chọn icon cho action tạo đường dẫn file PDF*

* Sync



*Hình 3.84. Action Sync*

* Action name ta đặt là Tạo đường dẫn file Phiếu NX.
* For a record of this table ta sẽ dùng dữ liệu từ bảng Phieu NX.
* Do this ta sẽ chọn App: go to another view within this app.
* Target ta đặt công thức như sau:



*Hình 3.85. Công thức đồng bộ ứng dụng*

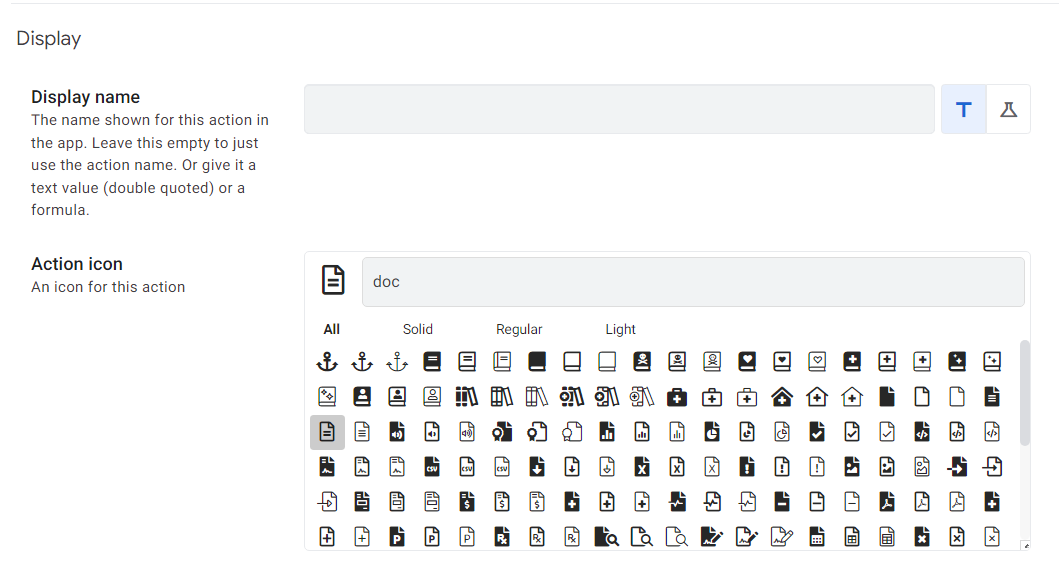
Công thức này sẽ tự động đồng bộ ứng dụng sau khi bấm tạo đường dẫn.

* Position ta chọn Hide vì action này sẽ không hiển thị trên ứng dụng.
* Mở phiếu NX



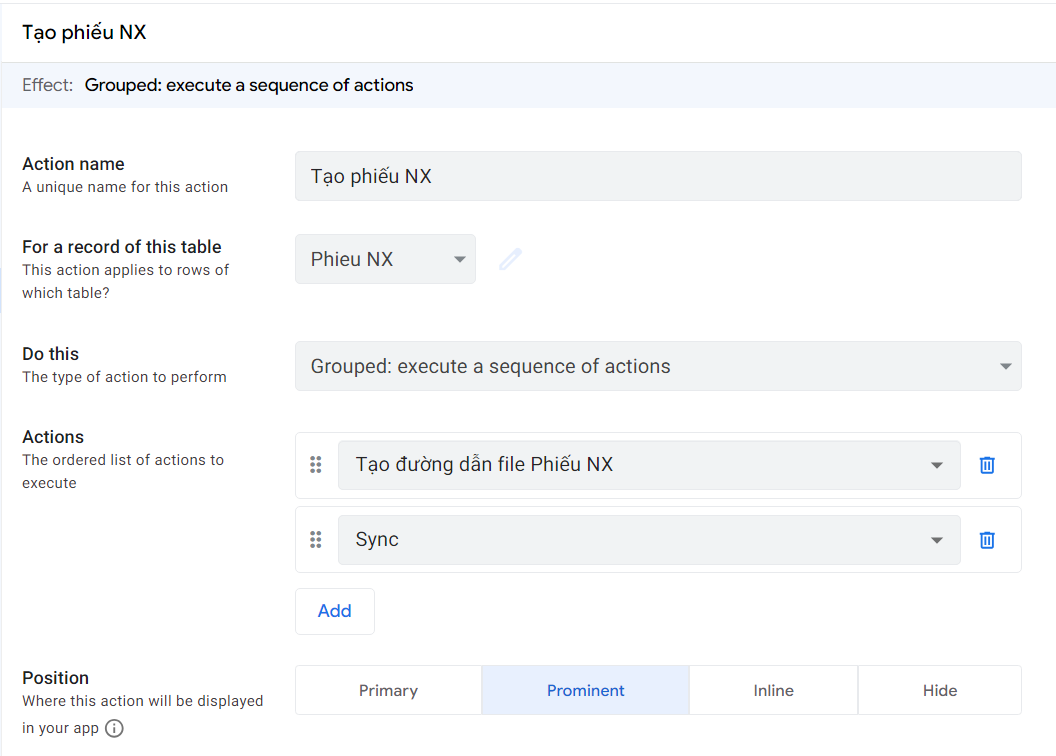
*Hình 3.86. Action Mở phiếu NX*

* Action name ta đặt là Mở phiếu NX.
* For a record of this table ta sẽ dùng dữ liệu từ bảng Phieu NX.
* Do this ta sẽ chọn External: open a file.
* File sẽ là [Đường dẫn file Phiếu NX].
* Position ta chọn Prominent.
* Vào mục Display > Action icon để chọn icon cho action.



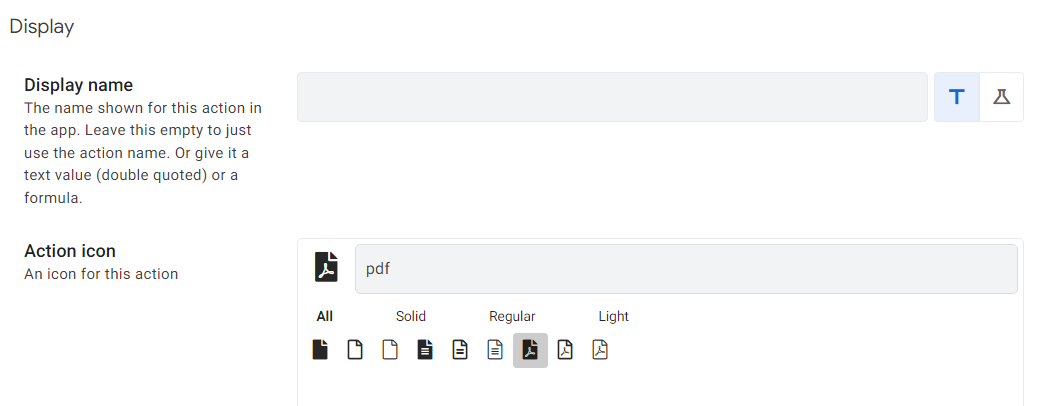
*Hình 3.87. Chọn icon cho action Mở phiếu NX*

* Tạo phiếu NX



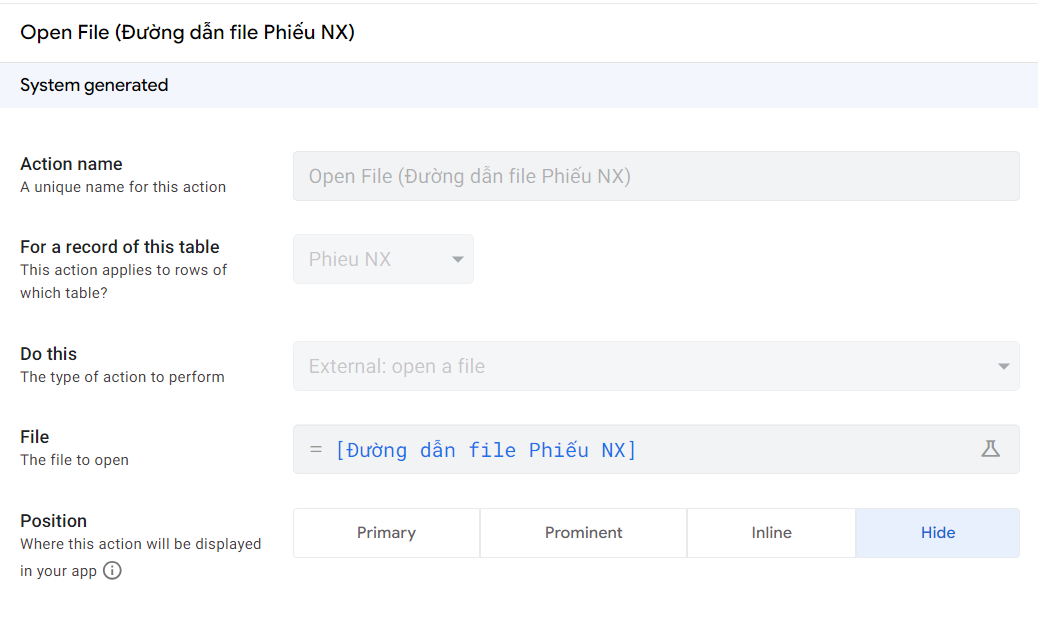
*Hình 3.88. Action Tạo phiếu NX*

* Action name ta đặt là Tạo phiếu NX.
* For a record of this table ta sẽ dùng dữ liệu từ bảng Phieu NX.
* Do this ta sẽ chọn Grouped: execute a sequence of actions.
* Actions sẽ Add 2 actions khác vào là Tạo đường dẫn file Phiếu NX và Sync.
* Position ta chọn Prominent.
* Vào mục Display > Action icon để chọn icon cho action.



*Hình 3.89. Chọn icon cho action Tạo phiếu NX*

* Open File (Đường dẫn file Phiếu NX)



*Hình 3.90. Action Open File*

Vì mục này Appsheet tạo sẵn nên ta không cần phải chỉnh sửa gì nhiều. Tại Position ta chọn Hide vì action này sẽ không hiển thị lên ứng dụng.

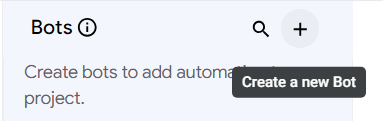
### **Tạo Automation**

Nếu đã đi đến bước này cơ bản ta sẽ tạo được 1 file PDF hoàn chỉnh như hình 3.6. Bây giờ hãy cùng nhau bắt đầu thực hiện nó.



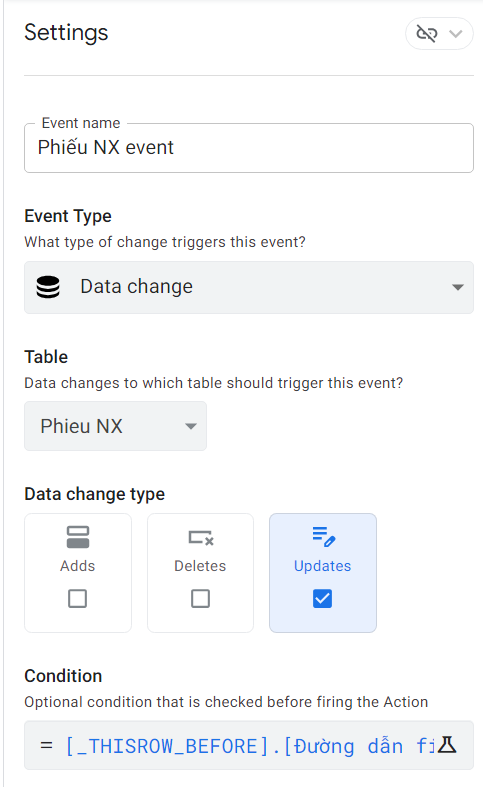
*Hình 3.91. Automation*

Đầu tiên ta sẽ tạo 1 bot mới



*Hình 3.92. Tạo bot mới*

Ở mục Configure event ta chọn Create a custom event sau đó chỉnh sửa như sau:



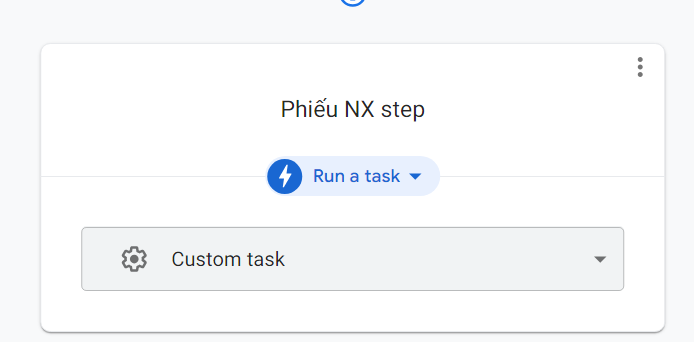
*Hình 3.93. Chỉnh sửa event cho bot*

* Event name ta đặt là Phiếu NX event.
* Event Type ta để như mặc định.
* Table ta chọn Phieu NX làm bảng dữ liệu chính.
* Data change type ta chọn Updates only.
* Condition ta nhập công thức



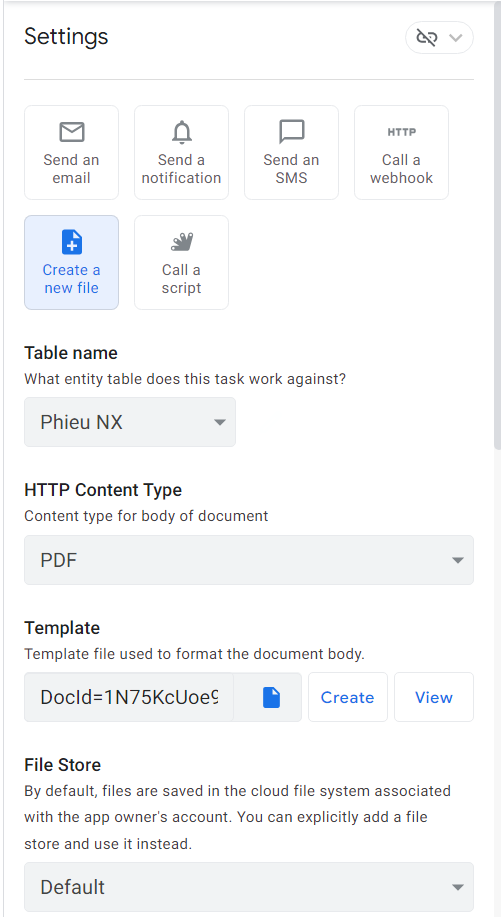
*Hình 3.94. Công thức bắt sự kiện*

Tiếp theo ta sẽ Add a step > Create a custom step để hoàn thiện bot. Ta đặt tên là Phiếu NX step.



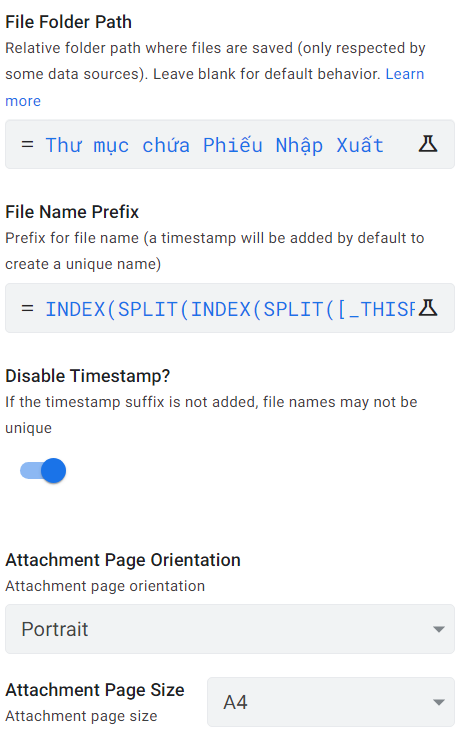
*Hình 3.95. Add a step*

Bước cuối cùng ta sẽ chỉnh sửa step này và lưu lại là bot của chúng ta sẽ có thể hoạt động được.



*Hình 3.96. Chỉnh sửa step của bot (1)*

* Chọn Create a new file.
* Table name ta chọn Phieu NX để sử dụng dữ liệu từ bảng Phieu NX.
* HTTP Content Type là PDF.
* Template nếu chưa có thì ta sẽ chọn new. Về phần template nhóm sẽ sử dụng file có sẵn.
* File Store ta để như mặc định.



*Hình 3.97. Chỉnh sửa step của bot (2)*

* File Folder Path ta để là Thư mục chứa Phiếu Nhập Xuất.
* File Name Prefix ta dùng công thức như sau:



*Hình 3.98. Công thức đặt tên file PDF*

* Attachment Page Orientation ta sẽ chọn Portrait để file được xuất ra theo chiều dọc.
* Attachment Page Size ta sẽ chọn là A4 theo tiêu chuẩn.

## **Chức năng kiểm kê**

### **Tổng hợp dữ liệu**

### **Tổng hợp dữ liệu của từng kho**

### **Kiểm kê**

## **Backup dữ liệu**

## **Demo ứng dụng**

# **Kết luận và hướng phát triển**

## **Kết luận**

Đề tài này đã tiến hành nghiên cứu nền tảng lập trình không code Appsheet. Trong đề tài, nhóm đã trình bày tổng quan về appsheet. Song song đó nhóm em cũng đã khảo sát, nghiên cứu về appsheet như: giới hạn của appsheet, các làm việc với dữ liệu trên appsheet, views, action,… Cuối cùng nhóm dùng kiến thức đã nghiên cứu được để thiết kế và tạo ra một ứng dụng dùng Appsheet làm nền tảng.

Trong quá trình thực hiện đề tài, Nhóm em đã sử dụng Google Sheets để làm cơ sở dữ liệu sau đó kết nối với Appsheet để tạo App. Bên cạnh đó em cũng cố gắng tham khảo các tài liệu liên quan bằng tiếng Việt lẫn tiếng Anh, tham gia vào các hội nhóm Appsheet để học hỏi các bậc tiền bối. Tuy nhiên do thời gian và trình độ có hạn nên không tránh khỏi những hạn chế và thiếu sót nhất định. Do vậy nhóm em thật sự mong muốn nhận được những góp ý cả về kiến thức chuyên môn lẫn cách trình bày.

## **Hướng phát triển**

Việc phát triển ứng dụng vẫn còn nhiều thiếu sót. Trong tương lai nhóm mong muốn cải thiện một số chỗ như:

* Thiết kế giao diện người dùng (UI) đơn giản hơn, dễ sử dụng hơn.
* Thêm một số chức năng khác hữu ích với người dùng hơn như liên kết với zalo để thông báo thông tin tồn kho.
* Cung cấp dịch vụ bảo trì, nâng cấp phần mềm định kỳ để đáp ứng nhu cầu thay đổi của cửa hàng nếu khách hàng có mong muốn thay đổi.

# **Tài liệu tham thảo**

|  |  |
| --- | --- |
| [1] | AppSheet Creators Community.[online]. <https://www.googlecloudcommunity.com/gc/AppSheet/ct-p/appsheet> |
| [2] | AppSheet Help.[online].  <https://support.google.com/appsheet?hl=en&sjid=3620578480876635593-AP#topic=11981018> |
| [3] | Hoà Data [Học Appsheet Online].[online].  <https://www.youtube.com/@hocdatacunghoa> |
| [4] | Học AppSheet.[online]. <https://www.youtube.com/@hocappsheet> |
| [5] | Nghiện Appsheet.[online]. <https://www.youtube.com/@NghienAppsheet> |